

Diseño Electrónico de la Tarjeta de Interfaz y Control para un Equipo de Seguimiento en Elevación en un Subsistema de Contramedidas Electrónicas

Autor: Laura Colero Álvarez

Director: M^a Luz Gil Heras

Tutor: César Sanz Álvaro

PÁGINA EN BLANCO INTENCIONADAMENTE

Proyecto Fin de Carrera

Resumen:

Como apoyo al Subsistema ECM tradicional de los sistemas de Guerra Electrónica, surge el Equipo de Seguimiento en Elevación cuyo fin es la búsqueda de la amenaza en elevación. Ante la necesidad de un mayor número de líneas entre los bloques internos de este Equipo de Seguimiento, se hace necesario el diseño de una Tarjeta de Interfaz y Control. La tarjeta se ocupará de realizar el control de datos y flujo de señales de módulos RF que componen el Equipo, haciendo de interfaz entre las Tarjetas Procesadoras y Digitalizadoras, así como entre la Tarjeta de Proceso y el Bloque de Recepción.

Una vez que se haya realizado el análisis de la necesidad de controlar estas señales, estudiado las especificaciones y los requisitos funcionales del Sistema, se procederá a implementar el diseño hardware y desarrollo firmware de la Tarjeta de Interfaz y Control de un Equipo de Seguimiento en Elevación para un Subsistema ECM, que constituye el objetivo de este PFC.

Abstract:

In order to find a threat on elevation, and as a support on the basic ECM subsystems of E-war systems, the Elevation Tracking Equipment appears. To the need of a bigger number of lines between inside blocks of this Tracking Equipment, the design of an Interface and Control Board is needed. This board will deal with the data control and with the flow of RF modules signals that set up the Equipment, and will also work as an interface between the Processing and Digitalizing Board, as well as between the Process Board and the Reception Block.

Once the signal control necessity analysis has been made, and once the specifications and functional System requirements have been studied, the hardware design and the firmware development will be ran as a target of PFC.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	11
1.1. Contexto Tecnológico.	11
1.2. Objetivo	15
1.3. Alcance.	17
2. REQUISITOS	19
2.1. Requisitos funcionales	19
2.1.1. Control de Módulos	19
2.1.2. Acciones	19
2.1.3. Accesos de la TIC	23
2.1.3.1. Acceso de la Tarjeta Procesadora	23
2.1.3.2. Acceso de la Tarjeta Digitalizadora	23
2.1.3.3. Acceso al Elementos de la UMA	24
2.1.3.4. Acceso al Resto de Unidades	25
2.2. Requisitos no funcionales	28
2.2.1. Emplazamiento físico	28
2.2.2. Alimentación de la tarjeta.	32
2.2.3. Condiciones ambientales. Temperatura y Humedad.	35
3. ANALISIS DE ESPECIFICACIONES TÉCNICAS	36
3.1. Estudio Preliminar del Diseño	36
3.1.1. Tratamiento de señales de control	38
3.1.1.1. Comunicaciones de la TIC del EAE	39
3.1.1.2. Control de Frecuencias	40
3.1.1.3. Control de Umbrales	42
3.1.1.4. Control de Atenuación	42
3.1.1.5. Control de Switches	45
3.1.1.6. Filtrado de Señales de control	47
3.1.2. Filtrado de Alimentaciones	47
3.1.3. Adaptación de Alimentaciones	48
3.2. Solución adoptada.	49
3.2.1. Interfaces de E/S en la TIC.	50
3.2.2. Bloques Hardware.	59
3.2.2.1. Dispositivo de Lógica Programable	59
3.2.2.2. Puerto Serie	66
3.2.2.3. Control de Umbrales	67
3.2.2.4. Control de Atenuaciones	69
3.2.2.5. Adaptación de Niveles	72
3.2.2.6. Filtrado de Alimentaciones	74
3.2.2.7. Adaptación de Tensiones	76
3.2.2.8. Otros	80
3.2.3. Plano Delineación Preliminar.	81
3.2.3.1. Conectores de la TIC	82
3.2.4. Lista Preliminar de Materiales	84

4. DISEÑO HARDWARE.	85
4.1. <i>Desarrollo de Diseño</i>	85
4.2. <i>Esquemas Eléctricos.</i>	86
4.2.1. Identificación de Librería Central	88
4.2.2. Captura de Esquemáticos	89
4.2.3. Verificación del esquemático	93
4.2.4. Compilación. CDB	93
4.2.5. Package.	95
4.3. <i>Lista de Materiales</i>	96
4.4. <i>Planos</i>	98
4.5. <i>Fabricación</i>	101
5. DISEÑO DE LA LÓGICA PROGRAMABLE	102
5.1. <i>Entradas/salidas de la FPGA.</i>	103
5.2. <i>Arquitectura de la FPGA.</i>	107
5.2.1. Comunicaciones.	108
5.2.2. Control de Frecuencias	110
5.2.2.1. Sintetizador de Sintonía	110
5.2.2.2. Filtro YIG	111
5.2.2.3. Sintetizador de Calibración.	112
5.2.3. Control de atenuaciones	115
5.2.4. Control de Umbrales	120
5.2.5. Control de Switches. Modo de Funcionamiento.	121
5.2.6. Gestión de Señales de Autocomprobación.	122
5.3. <i>Lenguaje VHDL.</i>	123
5.4. <i>Entorno de desarrollo.</i>	125
5.4.1. Creación de un Proyecto	129
5.4.2. Entrada de Diseño	132
5.4.3. Síntesis.	133
5.4.4. Asociación de Pines	135
5.4.5. Implementación	136
5.4.6. Fichero Programación de la FPGA	136
5.4.7. Fichero Programación de la PROM	137
5.5. <i>Diseño de los bloques funcionales.</i>	138
5.5.1. Control de Frecuencias	139
5.5.2. Control de Atenuaciones	141
5.5.3. Control de Umbrales	143
5.5.4. Control de Switches	145
5.5.5. Banco de Registros	146
5.5.6. Interfaz Puerto Serie	147
5.5.7. Fichero de mayor nivel. TOP.vhd	150
5.6. <i>Complejidad del Diseño.</i>	153
5.7. <i>Validación Funcional. Simulaciones</i>	155
5.7.1. Bloque "Registros.vhd"	157
5.7.2. Bloque "CONTROL_UMB.vhd"	158

5.7.3.	Bloque "CONTROL_ATEN.vhd"	158
5.7.4.	Bloque "CONTROL_SW.vhd"	158
5.7.5.	Módulo "CONTROL_FRECUENCIAS.vhd"	159
5.7.6.	Módulo "TOP.vhd"	159
6.	VERIFICACIÓN.	161
6.1.	<i>Pruebas de Aceptación</i>	161
6.2.	<i>Integración.</i>	163
7.	DOCUMENTACIÓN	166
8.	RESULTADOS Y CONCLUSIONES.	168
9.	PLANIFICACIÓN y PRESUPUESTO.	170
9.1.	<i>Planificación</i>	170
9.2.	<i>Costes Generales</i>	172
9.3.	<i>Análisis de Obsolescencia FPGA.</i>	174
10.	ANEXOS	179
11.	GLOSARIO DE TÉRMINOS Y ABREVIATURAS	180
12.	BIBLIOGRAFÍA	182

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE INTERCONEXIÓN ELEMENTOS DEL EAE	15
FIGURA 2: DIAGRAMA DE REQUISITOS FUNCIONALES	27
FIGURA 3 : MECÁNICA DEL EQUIPO DE APUNTAMIENTO EN ELEVACIÓN.....	28
FIGURA 4 : VISTA EN PERSPECTIVA DEL COFRE DEL EAE	29
FIGURA 5 : VISTA FRONTAL DISTRIBUCIÓN DE MÓDULOS DEL EQUIPO DE APUNTAMIENTO EN ELEVACIÓN	29
FIGURA 6 : MECÁNICA DEL MÓDULO UNIDAD DE MEZCLA Y ADAPTACIÓN	30
FIGURA 7 : VISTA SUPERIOR DE LA MECÁNICA QUE ALBERGA LA TARJETA INTERFAZ Y CONTROL	31
FIGURA 8 : ELEMENTOS CONTENIDOS EN LA UMA	33
FIGURA 9: ESQUEMA DE FILTRADO	34
FIGURA 10: PLANO DELINEACIÓN. ESPACIO TARJETA INTERFAZ Y CONTROL	36
FIGURA 11 : FORMAS DE ONDA RS-232	39
FIGURA 12 : CAPTURA DE BUS DE FRECUENCIA	40
FIGURA 13 : ATENUACIÓN EN TRES ETAPAS.....	43
FIGURA 14 : GRÁFICA FUNCIONAMIENTO DE HMC-C053	44
FIGURA 15 : DIRECCIONAMIENTO SEÑALES DE RF. CONTROL DE SWITCHES	45
FIGURA 16 : ADAPTACIÓN DE SEÑALES TTL-LVTTL	47
FIGURA 17 : ETAPA DE FILTRADO SEÑALES DIGITALES.....	47
FIGURA 18: ADAPTACIÓN DE ALIMENTACIONES	48
FIGURA 19 : BLOQUES FUNCIONALES HARDWARE DE LA TIC.....	49
FIGURA 20 : DIAGRAMA DE INTERCONEXIÓN DE LA TIC	50
FIGURA 21 : DIAGRAMA DE BLOQUES DE LA FAMILIA SPARTAN-II	60
FIGURA 22: CONFIGURACIÓN MASTER SERIAL	62
FIGURA 23 : SEÑALES DEL JTAG EN MODO MASTER SERIAL.....	62
FIGURA 24: RELOJ EXTERNO.....	63
FIGURA 25 : RED DE DISTRIBUCIÓN DEL RELOJ GLOBAL.....	63
FIGURA 26 : FORMA DE ONDA DE LA SALIDA DEL OSCILADOR	64
FIGURA 27 : CIRCUITO TÍPICO DE FUNCIONAMIENTO.....	66
FIGURA 28 : ARQUITECTURA DEL AD5290	67
FIGURA 29 : CRONOGRAMA SEÑALES AD5290	68
FIGURA 30 : CIRCUITO EQUIVALENTE AD5290.....	68
FIGURA 31 : DIAGRAMA CIRCUITO CONVERSOR DIGITAL-ANALÓGICO	70
FIGURA 32 : ESTRUCTURA INTERNA DE THS5671A	70
FIGURA 33 : PINOUT AD9754	71
FIGURA 34 : ESQUEMA ELÉCTRICO DE UN AMPLIFICADOR DE INSTRUMENTACIÓN	71
FIGURA 35 : ESQUEMA CONTROL DE ATENUACIONES.....	72
FIGURA 36 : PINOUT DEL CIRCUITO INTEGRADO 72LVX3245	73
FIGURA 37 : ADAPTACIÓN DE SEÑALES DE CONTROL	74
FIGURA 38 : FUNCIONAMIENTO DE LA SERIE BNX	75
FIGURA 39 : NUEVO BLOQUE PARA ADAPTACIÓN DE TENSIONES DIGITALES.....	76
FIGURA 40 : BLOQUE ADAPTACIÓN DE TENSIONES PARA LA UMA	77
FIGURA 41 : ADAPTACIÓN DE ALIMENTACIONES DIGITALES	77
FIGURA 42 : COLOCACIÓN DE CONDENSADORES DE DESACOPLO.....	80
FIGURA 43 : CONECTOR DIN 41612 96 PINES	82
FIGURA 44 : CONECTOR MICRO-D.....	82
FIGURA 45 : CONECTOR MMCX JACK MACHO RECTO PARA PCB	83
FIGURA 46 : TIRA DE PINES PARA PCB	83
FIGURA 47 : MIND MAP TAREAS DE DISEÑO HARDWARE	85
FIGURA 48 : ABOUT MENTOR DESIGN CAPTURE	86

FIGURA 49 : FLUJO DE DISEÑO DESIGN CAPTURE	87
FIGURA 50 : FLUJO DE DISEÑO EXPEDITION PCB	87
FIGURA 51 : IDENTIFICACIÓN LIBRERÍA CENTRAL.....	88
FIGURA 52 : WORKSPACE DE DESIGN CAPTURE.....	89
FIGURA 53 : VENTANA DE CREACIÓN DE NUEVO PROYECTO EN DESIGN CAPTURE	90
FIGURA 54 : VENTANA DIÁLOGO PARA NUEVO ESQUEMÁTICO	90
FIGURA 55 : MENÚ PLACEMENT	91
FIGURA 56 : PROPIEDADES DE LA LIBRERÍA DE UN COMPONENTE.....	91
FIGURA 57 : PROPIEDADES DE UNA NET.....	92
FIGURA 58 : SÍMBOLO CONECTOR INTERPÁGINA.....	92
FIGURA 59 : SÍMBOLO CONECTOR INTRAPÁGINA.....	93
FIGURA 60 : DIÁLOGO VERIFY.....	93
FIGURA 61: MENÚ COMPILE ICDB.....	94
FIGURA 62 : CUADRO INFERIOR DE COMANDOS.....	94
FIGURA 63 : MENÚ PACKAGE	95
FIGURA 64 : MENÚ EXTRACCIÓN DE LISTA DE MATERIALES.....	96
FIGURA 65 : DIÁLOGO EXTRACCION DE FICHERO *.BOM.....	97
FIGURA 66 : FICHERO *.BOM.....	97
FIGURA 67 : PLANO DELINEACIÓN	98
FIGURA 68 : CONECTOR DIN 41612 160 PINES	99
FIGURA 69 : PLANO DELINEACIÓN DEFINITIVO	100
FIGURA 70 : VENTANA EXPEDITION PCB	101
FIGURA 71 : ARQUITECTURA INTERNA DE LA FPGA.....	107
FIGURA 72 : ARQUITECTURA BANCOS DE REGISTROS.....	109
FIGURA 73 : BLOQUE INTERFAZ PUERTO SERIE.....	109
FIGURA 74 : BLOQUE CONVERSIÓN DE FRECUENCIA.....	110
FIGURA 75 : ESPECIFICACIÓN DE SEÑAL STROBE	111
FIGURA 76: BLOQUE CONTROL DE ATENUACIONES	119
FIGURA 77 : PROGRAMACIÓN DE UN POTENCIÓMETRO SPI.....	120
FIGURA 78 : ESQUEMA CONTROL DE MODOS DE FUNCIONAMIENTO DE LOS SWITCHES DEL EAE.....	121
FIGURA 79 : ESTRUCTURA DE LA DECLARACIÓN DE ENTIDAD	123
FIGURA 80 : ESTRUCTURA DEL CUERPO DE ARQUITECTURA	124
FIGURA 81 : FLUJO DE DISEÑO.....	126
FIGURA 82: WORKSPACE DE ISE PROJECT NAVIGATOR.....	129
FIGURA 83 : VENTANA DIÁLOGO CREACIÓN DE UN PROYECTO.....	129
FIGURA 84 : VENTANA DEVICE PROPERTIES	130
FIGURA 85 : VENTANA ELECCIÓN TIPO DE FICHERO	130
FIGURA 86 : RUTA CARPETA CON FICHEROS FUENTE	131
FIGURA 87 : VENTANA AÑADIR FICHEROS AL PROYECTO.....	132
FIGURA 88 : VENTANA SOURCES	133
FIGURA 89 : MENSAJE "CHECK SYNTAX"	134
FIGURA 90 : ESTADO DE VENTANA DE PROCESOS PARA SÍNTESIS	134
FIGURA 91 : ASOCIACIÓN DE PINES.....	135
FIGURA 92 : RELACIÓN DE PINES	136
FIGURA 93 : VENTANA PROCESSES TRAS LA GENERACIÓN DEL BITSTREAM	136
FIGURA 94 : GENERATE TARGET DEVICE	137
FIGURA 95 : BOUNDARY SCAN.....	137
FIGURA 96 : CONTROL DE FRECUENCIAS	139
FIGURA 97 : DECLARACIÓN DE ENTIDAD DE MÓDULO "CONTROL_FREQ"	139
FIGURA 98 : CUERPO DE ARQUITECTURA DE "CONTROL_FREQ"	140

FIGURA 99: CONTROL DE ATENUACIONES	141
FIGURA 100 : DECLARACIÓN DE ENTIDAD DE "CONTROL_ATEN"	141
FIGURA 101 : CUERPO DE ARQUITECTURA DE "CONTROL_ATEN"	142
FIGURA 102 : BLOQUE CONTROL DE UMBRALES	143
FIGURA 103 : DECLARACIÓN DE ENTIDAD DE "CONTROL_UMB"	143
FIGURA 104 : CUERPO DE AQUITECTURA DE "CONTROL_UMB"	144
FIGURA 105: BLOQUE CONTROL DE SWITCHES	145
FIGURA 106 : DECLARACIÓN DE ENTIDAD DE "CONTROL_SW"	145
FIGURA 107 : DECLARACIÓN DE ENTIDAD DE "REGISTROS"	147
FIGURA 108: BLOQUE INTERFAZ RS-232	148
FIGURA 109 : DECLARACIÓN DE ENTIDAD DE "RS-232_INTERFAZ"	148
FIGURA 110 : CUERPO DE ARQUITECTURA DE "RS-232_INTERFAZ"	149
FIGURA 111 : VENTANA SOURCES DEL PROYECTO "TIC_EAE"	150
FIGURA 112 : ESQUEMA RTL OBTENIDO DEL FICHERO "TOP.VHD"	151
FIGURA 113 : ESTRUCTURA INTERIOR DEL FICHERO "TOP.VHD"	152
FIGURA 114 : VENTANA "DEVICE SUMMARY"	154
FIGURA 115 : VENTANA SELECCIÓN FICHERO DE SIMULACIÓN	155
FIGURA 116 : ESTRUCTURA DE UN "TEST_BENCH"	156
FIGURA 117 : VENTANA DE PROCESOS PARA SIMULACIÓN	157
FIGURA 118 : IMAGEN DE LA TARJETA IMPLEMENTADA FÍSICAMENTE. CARA DE COMPONENTES	161
FIGURA 119 : VISTA DESPIECE BANDEJA INFERIOR.....	163
FIGURA 120 : MÓDULO UMA VISTA SUPERIOR.....	164
FIGURA 121 : MÓDULO UMA VISTA FRONTAL.....	164
FIGURA 122 : MÓDULO UMA VISTA TRASERA	165

ÍNDICE DE TABLAS

TABLA 1: MÓDULOS DE EAE (CYC)	19
TABLA 2: MÓDULOS DEL EAE (DPC)	19
TABLA 3 : ALIMENTACIONES DESDE LA TARJETA MADRE.....	32
TABLA 4 : TENSIONES REGULADAS	33
TABLA 5: CONSUMOS DE LOS ELEMENTOS DE LA TIC	34
TABLA 6 : DATO FRECUENCIA	41
TABLA 7 : DATO DESVIACIÓN.....	41
TABLA 8 : CÓDIGO FRECUENCIA.....	41
TABLA 9 : ELEMENTOS RF CON FRECUENCIA SINTONIZABLE	41
TABLA 10 : SEÑALES DE AUTOCOMPROBACIÓN DEL EAE	42
TABLA 11 : TABLA DE VERDAD MAADSS0013	43
TABLA 12 : SEÑALES DE AUTOCOMPROBACIÓN DE LA UMA.....	44
TABLA 13 : DESCRIPCIÓN DE LÍNEAS DE SEÑALES DE CONTROL	57
TABLA 14: DESCRIPCION SEÑALES DE LINEAS DE ALIMENTACION	58
TABLA 15 : RELACIÓN FPGA-PROM	61
TABLA 16 : MODOS DE CONFIGURACIÓN	61
TABLA 17: PINES DEDICADOS PARA SEÑALES DE RELOJ.....	64
TABLA 18: PINES DEDICADOS PARA SEÑALES DE ALIMENTACION	65
TABLA 19: ENCAPSULADOS DISPONIBLES DE LA FPGA	65
TABLA 20: CARACTERÍSTICAS ENCAPSULADOS MAX3314	66
TABLA 21: DESCRIPCION DE PINES DE 72LVX3245	73
TABLA 22: TABLA DE FUNCIONAMIENTO DEL 72LVX3245	73
TABLA 23 : REGULACIÓN DE TENSIONES	76
TABLA 24: CARACTERISTICAS DE LM117/LM317.....	78
TABLA 25 : OPCIONES DE ENCAPSULADOS PARA LM117/LM317A/LM317	78
TABLA 26: RELACIÓN DE CONSUMOS POR ALIMENTACIONES DE LA TIC	79
TABLA 27: RELACIÓN REGULADORES SEGÚN CONSUMOS.....	79
TABLA 28: LISTA PRELIMINAR DE MATERIALES	84
TABLA 29: INTERFAZ ENTRADAS/SALIDAS DE LA FPGA	106
TABLA 30 : RESOLUCIÓN DE DESVIACIÓN	113
TABLA 31 : TABLA DE VALORES DE DESVIACIÓN.....	113
TABLA 32 : RELACIÓN VALORES DE ATENUACIÓN	115
TABLA 33. RELACIÓN DE ATENUACIÓN Y CÓDIGO DEL DAC.....	117
TABLA 34. VALOR DE ATENUACIÓN TOTAL QUE SE INFORMARÁ A LA TARJETA DIGITALIZADORA.....	118
TABLA 35: HERRAMIENTAS DE DESARROLLO	125
TABLA 36: CABLES DE PROGRAMACION IMPACT.....	128
TABLA 37 : BANCO DE REGISTROS.....	146

PÁGINA EN BLANCO INTENCIONADAMENTE

1. INTRODUCCIÓN

La presente memoria describe el desarrollo del diseño electrónico de la Tarjeta de Interfaz y Control (TIC) para un Equipo de Apuntamiento en Elevación (EAE) de un Subsistema de Contramedidas Electrónicas (ECM). El desarrollo del diseño tiene lugar dentro del contexto de ejecución de proyectos de la empresa INDRA SISTEMAS y además se encuentra en el marco operativo de un proyecto militar con clasificación de RESERVADO. Por ello, y dada la difusión NO clasificada del presente documento, ciertos datos estratégicos e información son reservados y no son mostrados.

1.1. CONTEXTO TECNOLÓGICO.

Los Sistemas de Guerra Electrónica actuales se componen de dos Subsistemas principales denominados Subsistema ESM y Subsistema ECM.

Los **Subsistema ESM** (Electronic Support Measure) se encargan de vigilar la actividad electromagnética del ambiente en tiempo real y son capaces de construir un escenario de emisores de la zona de operación.

Los **Subsistema ECM** (Electronic Countermeasures) se encargan de realizar técnicas de engaño y perturbación radiando sobre los emisores detectados por el Subsistema ESM.

Los Subsistemas ECM actuales pertenecen a una nueva generación de subsistemas de contramedidas en banda radar, basados en la evolución de las técnicas de digitalización de alta velocidad respecto a sus predecesores. Incorporan los últimos avances en técnicas de digitalización rápida y proceso digital de señal, tanto en el dominio del tiempo como de la frecuencia. De esta manera, son capaces de explotar al máximo toda la potencia que le aportan dichas técnicas, redundando en una considerable versatilidad, así como en una resolución más precisa tanto en los parámetros temporales, como de frecuencia y fase.

Además, los subsistemas ECM presentan el aspecto fundamental de evolución futura, puesto que se ofrece una arquitectura modular fácilmente ampliable a partir del mismo núcleo de proceso y generación de técnicas.

Como apoyo al Subsistema ECM tradicional de los sistemas de Guerra Electrónica, nace el ***Equipo de Apuntamiento en Elevación (EAE)***, cuyo fin es la búsqueda y seguimiento mecánicos en elevación de la amenaza prioritaria que se desea contramedir. Este equipo aúna el conocimiento adquirido en proyectos anteriores desarrollados en INDRA SISTEMAS (recepción

digital, goniometría interferométrica y monopolso) para completar el diseño del Subsistema ECM.

El nuevo Equipo tiene como finalidad, por un lado, proporcionar las señales de radiofrecuencia necesarias para validar el ángulo de llegada en azimut del emisor a contramedir, y por otro lado, obtener ese ángulo en elevación para aquellas trazas que proceden de una plataforma aérea, realizando el seguimiento de la misma y reportando las medidas oportunas. De esta manera, el subsistema ECM podrá realizar contramedidas con seguimiento en la elevación de la dirección de la traza hostil.

El nuevo bloque se compone en las siguientes unidades:

- Antenas Receptoras (ARX), constituidas por dos arrays de antenas combinadas en horizontal y vertical.
- Calibrado y Conversión (CYC), que se encarga de la conversión de señales de radiofrecuencia (RF) a Frecuencia Intermedia (FI) y de la calibración del Equipo de Apuntamiento en Elevación.
- Bloque de Digitalización, Proceso y Control (DPC), que además de contener la **Tarjeta de Interfaz y Control (TIC)** objeto de este proyecto, incluye un Módulo de Digitalización y un Módulo de Proceso.

El bloque de las **Antenas Receptoras (ARX)**, es una unidad que está constituida por dos arrays de antenas espirales, combinados en horizontal y vertical, que presenta una doble funcionalidad:

- Con el array horizontal (AH) y partiendo de la información base que proporcionará el Subsistema ESM, se obtienen las señales de radiofrecuencia necesarias para validar el ángulo de llegada en azimut del emisor a contramedir y se preapunta el EAE en azimut.
- Las señales proporcionadas por el array vertical (AV) se utilizan exclusivamente para la medida de la elevación de la amenaza de interés y el seguimiento de la misma en dicha dimensión (combinando técnicas monopolso e interferométrica). Dentro de este conjunto, se incluyen switches para poder introducir señal de RF en las cadenas de medida y realizar tareas de autocalibración en amplitud y fase.

El **Calibrado y Conversión (CYC)**, en líneas generales consta de las siguientes unidades, para las que además se detallan los elementos que requieren control:

- a. Unidad de Filtrado y Amplificación (UFA)
 - Filtro YIG
 - Transfer Switch
 - Switches
 - Detectores de Nivel
- b. Unidad Oscilador Local (UOL)
 - Oscilador de Sintonía (Para conversión a Frecuencias Intermedias)
 - Detector de Nivel
- c. Unidad de Calibración (UC)
 - Oscilador de Sintonía (Para calibración de EAE)
 - Detector de Nivel
 - Switch de Calibración
- d. Unidad de Mezcla y Adaptación (UMA)
 - Atenuadores y Ecualizadores
 - Detectores de Nivel
- e. Conjunto Electromecánico de Ensamblajes de antenas.
 - Control Mecánico del posicionamiento de Antenas

El **Bloque de Digitalización, Proceso y Control (DPC)**, tiene las siguientes funciones:

- Dar soporte físico al software de digitalización y proceso de las señales recibidas, así como al software de control del EAE (incluido el posicionamiento de las antenas).
- Dar soporte físico a todas las interfaces de comunicaciones entre el EAE y el resto del sistema

Tal y como se ha mencionado antes, desde el punto de vista funcional, este bloque está formado por los siguientes elementos:

- **Módulo de Digitalización**, con la Tarjeta Digitalizadora que se ocupa de obtener las muestras de las señales adaptadas que procedan del array vertical. Esta tarjeta se ocupa de realizar la digitalización de las señales generando así los descriptores de pulsos y de detecciones que serán enviados al módulo de Proceso.
- **Módulo de Proceso**, dentro de éste se encuentra la Tarjeta Procesadora que se encarga de recibir los descriptores procedentes del Módulo de Digitalización y posee la capacidad hardware de filtrarlos y almacenarlos en buffers. Lleva a cabo el desentramado de los mismos y caracteriza los parámetros del que se identifique como amenaza prioritaria.
- **Tarjeta Interfaz y Control del EAE (TIC del EAE)**: esta tarjeta, objetivo de este Proyecto Fin de Carrera, se ocupa de proporcionar interconexión complementaria, tanto física como funcional, entre ciertos elementos del EAE y así extender el control sobre éstos de la Tarjeta Procesadora y Tarjeta Digitalizadora.

1.2. OBJETIVO

El objetivo de este proyecto es el diseño, fabricación y programación de la **Tarjeta de Interfaz y Control (TIC)**, cuyo propósito es el de dar solución a la necesidad de conexión física y funcional entre los distintos bloques que constituyen el Equipo de Apuntamiento en Elevación (EAE), formando parte así de un Subsistema ECM.

En otras palabras, la finalidad de este Proyecto Fin de Carrera se fundamenta en la necesidad de gestionar y monitorizar cierto número de líneas de control entre la Tarjeta de Proceso y el Bloque CYC, así como entre la Tarjeta de Proceso y la Tarjeta de Digitalización, y entre el Bloque CYC y las Antenas Receptoras (ARX). De esta manera, aparece el concepto de tarjeta interfaz y de control de los módulos del Equipo de Apuntamiento en Elevación.

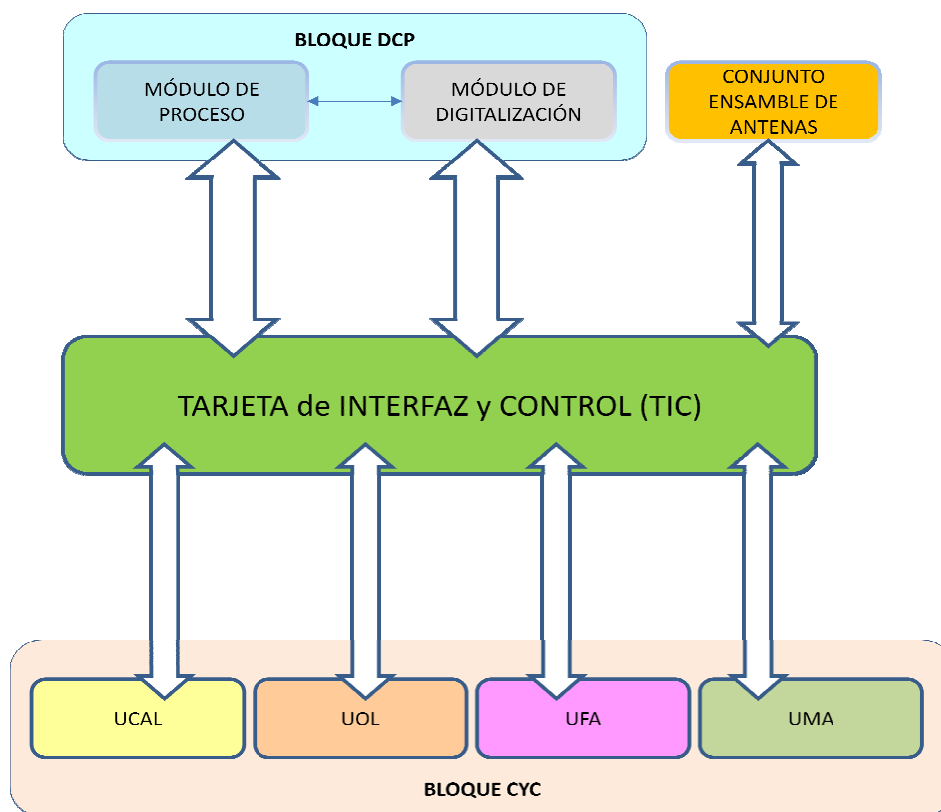


FIGURA 1: DIAGRAMA DE INTERCONEXIÓN ELEMENTOS DEL EAE

El contexto operativo de la TIC incluye aplicar etapas de amplificación a las señales de RF procedentes de las antenas del array vertical, controlar la mezcla para su conversión a FI (frecuencias intermedias) a partir de las señales de osciladores locales y gestionar la adaptación de las señales FI en ancho de banda y nivel para su posterior digitalización. La TIC responde por tanto a la necesidad de ampliar las líneas de control que proporciona el módulo

de proceso, y basará su diseño en la utilización de una FPGA (Field Programmable Gate Array) para dicho propósito.

Por requisitos del diseño en conjunto del EAE, la TIC se ubicará físicamente dentro de la Unidad de Mezcla y Adaptación (UMA), y por ello, el diseño de la tarjeta obedecerá no sólo a ejercer funciones de control sobre los módulos que componen el EAE, sino además a filtrar alimentaciones y tratamiento de los datos que se van a distribuir a la UMA.

Por tanto, el objetivo de este Proyecto Fin de Carrera consiste en el diseño hardware de una Tarjeta Interfaz y de Control y desarrollo firmware de la FPGA albergada en la misma que, basándose en las especificaciones, requisitos funcionales y limitaciones del sistema, se encargue de gestionar ciertas señales cuyo interfaz inicial es el siguiente:

- Capturar el dato de frecuencia de sintonía procedente de la Tarjeta de Proceso (TP), con el fin de adaptarla y multiplexarla hacia los módulos:
 - UFA: Unidad de Filtrado y Amplificación
 - UOL: Unidad Oscilador Local.
 - UC: Unidad de Calibración.
- Hacer posible el control sobre cada uno de los switches del Equipo atendiendo al modo de funcionamiento que venga marcado desde la Tarjeta de Proceso (TP)
- Filtrar alimentaciones que necesitarán los elementos de la UMA, así como permitir el manejo desde la TP de los atenuadores y ecualizadores de la misma.
- Informar a la Tarjeta Digitalizadora (TD) en tiempo real, de cuál es el estado de los atenuadores y ecualizadores programados por la TP.

En definitiva, para este diseño se parte del entorno operativo de la TIC, cuyos aspectos vendrán definidos como especificaciones, requisitos y limitaciones del sistema, y que se detallará más adelante.

1.3. ALCANCE.

Para resolver la problemática planteada en el apartado anterior, y abordar el diseño de la Tarjeta de Interfaz y Control (TIC), se detallan a continuación los trabajos y acciones que se abarcarán en este proyecto.

La primera fase de diseño de este Proyecto Fin de Carrera consistirá en enmarcar el contexto de trabajo de la TIC y en realizar un estudio funcional de las condiciones necesarias que debe cubrir esta tarjeta con su papel en el EAE. Es decir, se desglosarán las tareas que debe efectuar la TIC y se debe estudiar qué tipo de función cumple y qué tipo de líneas son las que pueden cubrir esta funcionalidad.

Además será necesario realizar un análisis de los requisitos relacionados con las limitaciones como alimentaciones, consumos y demás elementos que puedan condicionar o influir en las decisiones de diseño que se pueden tomar.

Después de examinar estos aspectos elementales, se procederá a plantear las posibles alternativas que se pueden llevar a cabo para cada una de las funcionalidades requeridas para este diseño.

Finalmente, una vez recopilados todos los datos y la información que define esta tarjeta, se procederá a diseñar el hardware de la misma atendiendo a la selección de componentes y anexando esquemáticos y ficheros BOM (*Bill of materials*).

En este Proyecto de Fin de Carrera se pretende trabajar con una FPGA para llevar a cabo parte de la operativa de control que debe dirigir la TIC. Por lo tanto, el siguiente paso en este Proyecto Fin de Carrera, consistirá en desarrollar el diseño que se integrará en la FPGA (el cual se realizará en lenguaje VHDL) definido en la etapa de diseño hardware anterior, explicando la arquitectura y las interfaces de entradas/salidas.

Después de esta fase se procederá a verificar el correcto funcionamiento del diseño mediante simulaciones que permitan comprobar que la unidad en el sistema va a cumplir con sus requisitos funcionales.

Para finalizar este Proyecto Fin de Carrera, se expondrán resultados y conclusiones recopiladas una vez la tarjeta esté operando dentro de su contexto de trabajo.

Todo este desarrollo del diseño así como el resto de fases mencionadas, se realizará en el marco de trabajo de **INDRA SISTEMAS S.A**, condicionándose así en algunos aspectos la toma de decisiones adecuándose a la metodología de trabajo de la empresa.

En definitiva, los propósitos que se pretenden alcanzar en este Proyecto Fin de Carrera son a grandes rasgos los siguientes:

- Desarrollar un proyecto bajo la metodología de diseño profesional y en el contexto operativo de la empresa INDRA SISTEMAS S.A.
- Familiarización del proceso de desarrollo de un proyecto de hardware digital, desde el planteamiento inicial hasta su fabricación, montaje y funcionamiento final en su contexto de trabajo.
- Análisis de las especificaciones, requisitos y limitaciones del contexto operativo para implementar el diseño más óptimo
- Practicar la realización de los diversos cálculos para la implementación del diseño electrónico que realice las funciones motivo de este Proyecto Fin de Carrera.
- Ejercer la búsqueda de diferentes componentes electrónicos de distintos fabricantes (circuitos integrados, conectores...) y realizar la selección según se ajusten a las necesidades del proyecto. Entre dichos componentes, cabe destacar que una de las tareas más amplia será el estudio de determinadas FPGAs de Xilinx para llevar a cabo la programación del diseño funcional.
- Mostrar la metodología de diseño, desarrollo y documentación de una PCB, y modelado, simulación y testeo de sistemas digitales en VHDL.
- Investigar estándares de transmisión de datos como RS-232 y SPI.
- Análisis de conocimientos respecto a dispositivos lógicos programables en cuanto a su funcionamiento y programación, y de los accesorios disponibles que puedan ser necesarios como programadores externos, software específico...
- Ejercer la preparación de resultados y conclusiones obtenidas durante la elaboración de este Proyecto Fin de Carrera, así como llevar a cabo una reflexión sobre las posibilidades de mejoras o ampliaciones.

2. REQUISITOS

2.1. REQUISITOS FUNCIONALES

En este apartado se analiza el comportamiento interno y funcionalidades específicas que marcarán el objetivo final del diseño.

La Tarjeta de Interfaz y Control (TIC) del EAE, objeto de este proyecto, debe tener como requisito principal la gestión del control de los módulos que componen el EAE. El sistema plantea todas estas tareas como requisitos funcionales de la TIC del EAE.

2.1.1. CONTROL DE MÓDULOS

Basándose en la figura 1, se comenzará detallando los módulos sobre los que va a ejercer control esta tarjeta, que son los siguientes:

MÓDULOS CYC
Unidad de Filtrado y Amplificación (UFA)
Unidad de Mezcla y Adaptación (UMA)
Unidad de Oscilador Local (UOL)
Unidad de Calibración (UC)
Conjunto de Ensamble de Antenas

TABLA 1: MÓDULOS DE EAE (CYC)

Y además tendrá que realizar acciones de interfaz con los módulos:

MÓDULOS DPC
Tarjeta Procesadora
Tarjeta Digitalizadora

TABLA 2: MÓDULOS DEL EAE (DPC)

2.1.2. ACCIONES

En este punto se describe de un modo resumido el funcionamiento en general de la TIC, partiendo de que las acciones de esta tarjeta están comandadas por la Tarjeta Procesadora (TP) y al mismo tiempo se le comunica esta información de funcionamiento a la Tarjeta Digitalizadora (TD).

Como requisito prioritario se establece que el control que debe ejercer la TIC sobre estos módulos detallados en el punto anterior se basa en las siguientes **acciones**:

- CONTROL DE FRECUENCIA:

Se determina que el sistema funcionará en un rango entre dos frecuencias ($F1$ y $F2$), con una resolución de $B1$ (Hz). Así mismo es requisito un valor de Desviación de Frecuencia (DF), necesario para Calibración del Equipo de Apuntamiento en Elevación (EAE). El valor de Desviación se recibe de la Tarjeta Procesadora (TP) y deberá tener una resolución denominada $B2$ y el rango establecido será de $\pm DF$. En definitiva, en la Calibración del EAE, la TP proporcionará información de la frecuencia junto con un valor de desviación.

Este control se realizará atendiendo al valor programado por la TP y se distribuirá adecuadamente a los módulos de CYC.

- CONTROL DE ATENUACIONES:

Debe hacer posible que se introduzca una atenuación variable cuyo valor será controlado externamente por la TP para cada canal, y además proporcionará información sobre su autocomprobación.

Como requisito se indica que la atenuación se realiza en tres etapas situadas en diferentes puntos de la cadena de RF. Se señala que, para conseguir esto, los elementos que debe controlar la TIC en la UMA serán:

- Atenuador de RF (HMC-C053)
- Atenuador de FI (MAADSS0013)
- Atenuador de Ecualización (MAADSS0013)

Todo este proceso tiene su origen en la Tarjeta Procesadora que se ocupa de marcar las pautas de atenuación. Pero la TP no dispone de líneas físicas suficientes para ocuparse de ello, por lo que se apoya en la TIC y a través de ella se encargará de la distribución de la atenuación para el EAE.

Finalmente el valor total de atenuación será compuesto por el resultado del valor de los tres elementos y será comunicado continuamente a la Tarjeta Digitalizadora.

La Unidad de Mezcla y Adaptación debe realizar la atenuación de forma paralela sobre dos canales independientes del AV.

En la TIC deberán existir también señales para procesos de autocomprobación denominadas señales de BITE. La TIC debe tomar ese valor de los Detectores de Nivel y ejercer de medio interfaz hacia la TP, que es quien debe procesar este valor de BITE. Dado que la detección de nivel en banda no es plana, se necesita conmutar umbrales en función de la frecuencia. Por lo tanto, queda como tarea para la TIC generar valores umbrales configurables para cada uno de los detectores de nivel que se encuentran en la UMA. Los valores vendrán determinados por tablas de funcionamiento.

Se determina como requisito que se cuenta con una señal denominada OVERLOAD, que se indicará con un nivel alto cuando supere los $+A_1$ dB y no se recupere hasta que no baje de $+A_2$ dB, el sistema deberá ajustarlo y para ello la TIC del EAE debe servir de interfaz entre los elementos de la UMA y la TP.

- CONTROL DE SWITCHES

Mediante el posicionamiento de ciertos switches del EAE se realiza la conmutación y encaminamiento de las señales de RF del Conjunto de Ensamble de Antenas. Las señales de control de estos switches son necesarias para la correcta recepción y transmisión de las señales de RF, así como direccionamiento de señales de autocomprobación y calibración del EAE.

En relación a este Control de Switches del sistema, la TIC debe ocuparse de accionar cada uno de ellos para manejar el modo de funcionamiento que vendrá determinado desde la Tarjeta de Proceso.

Se describen dos modos de funcionamiento:

- Modo de Calibración: mediante una señal de calibración controlada externamente se debe permitir la caracterización de las antenas de seguimiento contrastando valores y corrigiéndolos. La calibración se realizará en todo el rango de frecuencias de funcionamiento de EAE.
- Modo de Antena: se trata del modo de estado operativo normal del sistema. En cuanto a los canales del AH, se deja pasar la señal de las antenas hasta el sistema ECM para validar en azimut la señal prioritaria. En cuanto a los canales del AV, se deja pasar la señal de las antenas hasta el modulo DCP para que ésta sea digitalizada y procesada y hacer el seguimiento oportuno

Los switches que hay que accionar para llevar a cabo este control son los siguientes:

- SWITCH_ANTENA: Está ubicado en el Conjunto de Ensamble de Antenas. Permite la entrada de la señal de RF desde el Array Vertical, o bien conmuta al estado de calibración del mismo.
- SWITCH_ECM: Está situado en el módulo Unidad de Filtrado y Adaptación. Permite la entrada de la señal de RF desde el Array Horizontal hacia el ECM, o bien conmuta al estado de calibración del mismo.
- SWITCH_CALIBRACIÓN: Esta situado en el módulo Unidad de Calibración. Su uso es sólo aplicable en el Modo de Calibración, según su posición calibra la cadena del EAE o la cadena de ECM.
- SWITCH_FILTRO: Está situado en el módulo UFA. Sirve para eliminar una banda de RF, si el sistema lo precisa, en el Modo Antena.

2.1.3. ACCESOS DE LA TIC

Una vez destacados en puntos anteriores los aspectos principales sobre las acciones que debe ejercer la TIC sobre los módulos de EAE, se mencionarán ahora los accesos con los que cuenta la TIC para realizar dicho control.

2.1.3.1. ACCESO DE LA TARJETA PROCESADORA

Como se ha mencionado anteriormente, las acciones que debe llevar a cabo la TIC vienen determinadas bajo la orden de la Tarjeta Procesadora. Por lo tanto la TIC debe atender a las interfaces con la TP que son las siguientes:

- Bus de 16 señales TTL de entrada hacia la TIC

Este bus será la vía a través de la cual se realiza el **Control de Frecuencia** en el EAE, mencionado en el punto 2.1.2

- Puerto serie RS-232

Existe un interfaz de comunicaciones con la TP, que servirá para que la TIC del EAE y la TP puedan intercambiar información relevante al control sobre los atenuadores y los switches, que conformarán el funcionamiento del sistema. Se trata de un puerto serie RS-232. Este puerto se sigue empleando porque son sencillos, baratos y permiten la interoperabilidad entre dispositivos. En este caso, para este Proyecto Fin de Carrera, este puerto permite pasar información de la TP hacia la TIC y viceversa, mediante solamente tres hilos físicos entre ambas tarjetas: TX, RX y GND.

- 1 señal digital (OVERLOAD) de salida desde la TIC

Durante todo el tiempo, es necesario que la TP conozca el estado de esta señal, que será obtenida desde los módulos de la UMA, y adecuada y pasada a la TP para que según su estado opere con los atenuadores de la manera que considere oportuna.

Estos accesos serán conexiones físicas entre la tarjeta madre y la TIC

2.1.3.2. ACCESO DE LA TARJETA DIGITALIZADORA

Así mismo se define una comunicación entre la TIC y la Tarjeta Digitalizadora (TD) para transmitirle a ésta información en tiempo real de la programación de los atenuadores. Los accesos que se especifican para permitir la conexión con la TD son:

- Bus de 11 señales TTL de salida desde la TIC

Contiene la información sobre la atenuación. Se trata del valor total de atenuación que será compuesto por el resultado del valor de las tres etapas.

- 1 señal digital (BLANKING) de entrada hacia la TIC

Desde el cofre del EAE, y procedente del subsistema ECM, se determina que existe una línea que sirve para señalar al Subsistema ESM que el sistema está ejecutando contramedidas, con el fin de prevenir que la contramedida sea detectada erróneamente como una amenaza.

La TIC debe ocuparse de recoger este valor y actuar consecuentemente. Estos accesos serán conexiones físicas entre la tarjeta madre y la TIC

2.1.3.3. ACCESO AL ELEMENTOS DE LA UMA

En este módulo la TIC realizará el **Control de Atenuaciones**. Como requisito se indica que la atenuación se realiza en tres etapas situadas en diferentes puntos de la cadena de RF. Se señala que, para conseguir esto, los elementos que debe controlar la TIC en la UMA son:

- Atenuador de RF (HMC-C053): se dispondrá de 30 dB de atenuación, que deberá ser controlado con una línea de tensión de control (V_{CONTROL}) entre 0 y -3V. Se necesita una resolución de 2 dB.
- Atenuador de FI (MAADSS0013): se dispondrá de 15dB de atenuación en pasos de 1 dB controlado por un bus 4 señales digitales (AT3-AT0).
- Atenuador de Ecualización (MAADSS0013): se dispone de 15dB de atenuación en pasos de 1 dB controlado por un bus de 4 señales digitales (AT3-AT0).

A partir de esto, se puede vislumbrar ya qué accesos existirán entre los atenuadores de la UMA y la TIC.

Se detalla, como requisito, que en el Control de la Atenuación se parte de una situación de “X” dB de atenuación que será función de la frecuencia de sintonía del sistema y en función del nivel de la señal se modificará la atenuación. Como ya se ha mencionado este proceso está gobernado desde la Tarjeta Procesadora (TP) marcando las pautas de atenuación, pero debe ser controlado a través de la TIC, realizando así su función de tarjeta de interfaz y de control, quien se ocupe de la distribución de la atenuación para el EAE.

Finalmente el valor total de atenuación será compuesto por el resultado del valor de las tres etapas y será comunicado continuamente mediante el bus de 11 líneas a la Tarjeta Digitalizadora ya mencionado.

Por lo tanto, en cuanto a los atenuadores tendrá que tener las siguientes líneas de control:

- 4 líneas TTL para los atenuadores MAADSS0013 de la etapa ECU por cada canal.
- 4 líneas TTL para los atenuadores MAADSS0013 de la etapa FI por cada canal.
- 1 línea analógica entre 0 y -3V para el atenuador HMC-C053 para cada canal.

La TIC tendrá que realizar además el tratamiento de las siguientes señales:

- 2 señales TTL que indiquen el resultado de autocomprobación (BITE) de la unidad para cada uno de los canales.
- 2 línea de tensión umbral (V_{umbral}) para dos Detectores de Nivel.
- 2 líneas TTL de indicación de sobrecarga (OVERLOAD), una por canal.

Las señales de los módulos de atenuación, excepto la línea analógica de 0-3V, entrarán a ellos mediante conectores Micro-D de 15 pines (uno por canal), junto con las señales de autocomprobación. La señal de sobrecarga vendrá mediante cable coaxial. Por ello habrá que contemplar los conectores adecuados en la PCB que reciban/transmitan estas señales.

2.1.3.4. ACCESO AL RESTO DE UNIDADES

La TIC se ocupa de llevar a cabo el tratamiento, filtrado y control de las señales de los módulos del EAE. Las dos tareas fundamentales son la de conducirle adecuadamente la frecuencia de sintonía del sistema a cada modulo, y accionar todos los switches para que se comporten según el modo de funcionamiento.

Los accesos a dichos módulos que necesitan que les indiquen la frecuencia de sintonía son los siguientes:

- **Modulo Unidad de Filtrado y Amplificación**
 - Bus de 12 señales TTL para frecuencia del filtro YIG (FREQ_YIG[11...0]).
 - 1 señal de STROBE, para marcar que hay dato válido en las salidas anteriores.
 - 2 señales de test para procesos de comprobación de esta unidad (BITE_UFA0, BITE_UFA1), procedente de los Detectores de Nivel.

- **Módulo Unidad Oscilador Local**

- Bus de 20 señales TTL para frecuencia de sintonía del EAE (FREQ_SINT[19...0])
- 1 señal de STROBE para marcar que hay dato válido en las salidas anteriores.
- 1 señal de test para procesos de comprobación de esta unidad (BITE_SINT).

- **Módulo Unidad de Calibración**

- Bus de 15 bits para la frecuencia de calibración del EAE (FREQ_CAL[14...0])
- 1 señal de STROBE para marcar que hay dato válido en las salidas anteriores.
- 1 señal de test para procesos de comprobación de la unidad (BITE_CAL).

En relación al **Control de Switches** del sistema, el modulo UMA a través de la TIC debe ocuparse de accionar cada uno de ellos para manejar el modo de funcionamiento del EAE que vendrá determinado desde la Tarjeta de Proceso.:

- 2 líneas TTL como señales de control para el switch de selección ANTENA
- 2 líneas TTL como señales de control para el switch de selección CALIBRACIÓN
- 2 líneas TTL como señales de control para el switch de selección ECM
- 1 línea TTL como señal de control para el switch que activa el FILTRO del interior de la UFA

Todo este tránsito de información sobre las señales de control de los switches se realizará mediante conexiones a la tarjeta madre del cofre desde la propia TIC.

Finalmente, para sintetizar el comportamiento y funciones específicas que debe desempeñar la Tarjeta de Interfaz y Control, y para realizar un posterior diseño se agrupan y organizan todos los requisitos funcionales mediante un mind map que contenga la información detallada y simplifique su manejo. En la siguiente figura podemos observar dicho mapa:

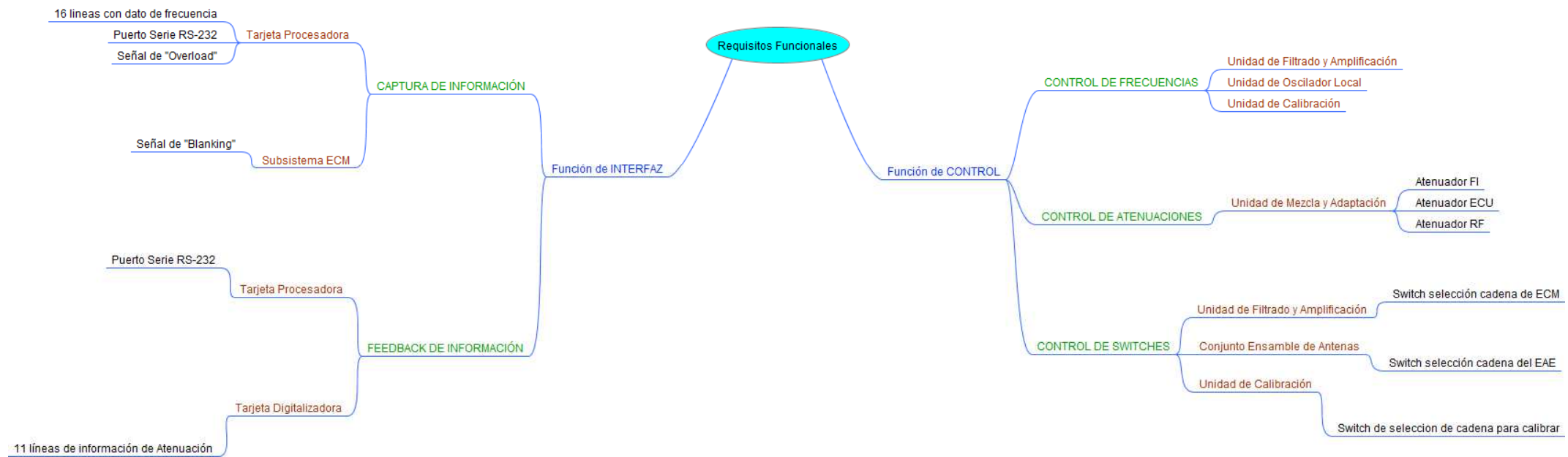


FIGURA 2: DIAGRAMA DE REQUISITOS FUNCIONALES

2.2. REQUISITOS NO FUNCIONALES

En este apartado se destacan aquellos aspectos que condicionan el diseño de la TIC independientemente del comportamiento interno de la TIC en sí.

Se realiza un análisis de las condiciones externas a la TIC que influyen en la toma de decisiones en la implementación del diseño.

2.2.1. EMPLAZAMIENTO FÍSICO

En primer lugar se debe tener en cuenta la situación física de la TIC en el EAE, así como las dimensiones que vienen determinadas ya para diseñar la PCB soporte de la TIC.

Como requisito viene determinado que, debido a que una gran parte de las tareas de la TIC van enfocadas hacia cubrir necesidades en la Unidad de Mezcla y Adaptación (UMA), la tarjeta en sí se ubicará físicamente como parte de la UMA, albergándose dentro del propio modulo. Se debe definir, por tanto una interfaz de entrada/salida en este módulo para que los accesos al resto de elementos del EAE sean posibles.

En primer lugar se analiza la mecánica del EAE, cuya distribución es la siguiente:

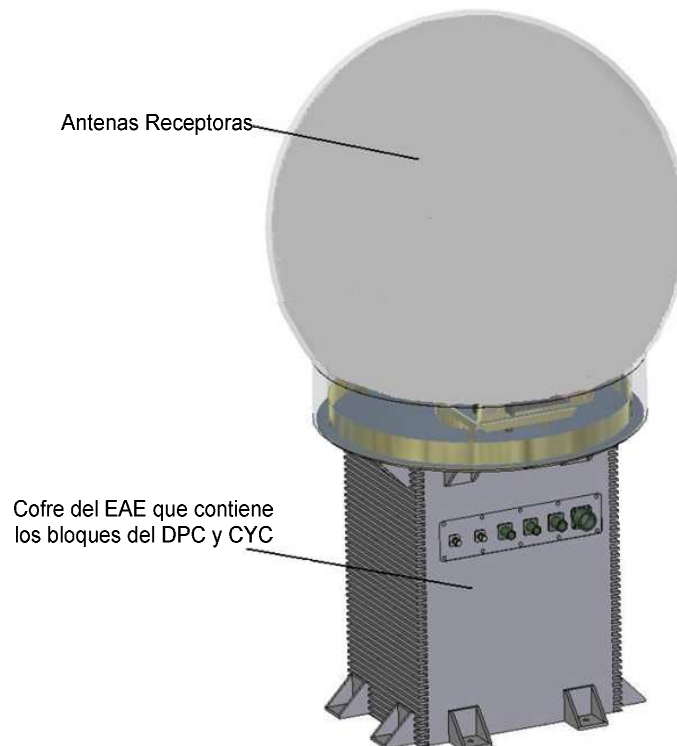


FIGURA 3 : MECÁNICA DEL EQUIPO DE APUNTAMIENTO EN ELEVACIÓN

Por lo tanto la Tarjeta Interfaz y de Control, motivo de este proyecto fin de carrera, se albergará dentro del Módulo de la UMA en el interior del cofre del EAE.

La distribución interior de los módulos dentro del cofre viene marcada de la siguiente manera:

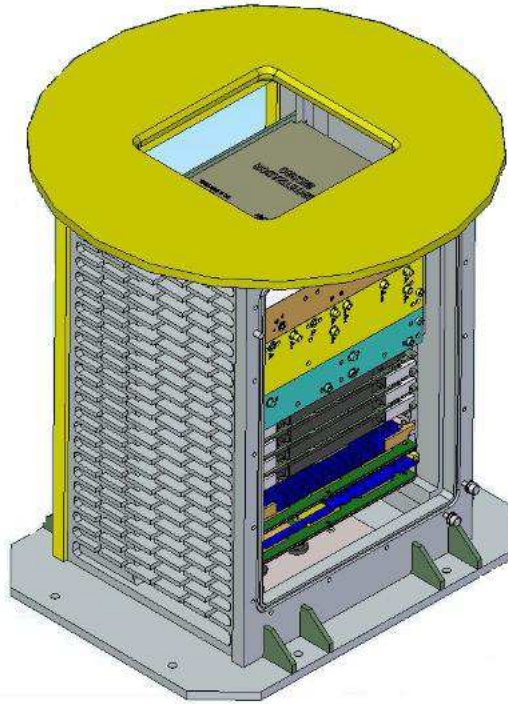


FIGURA 4 : VISTA EN PERSPECTIVA DEL COFRE DEL EAE

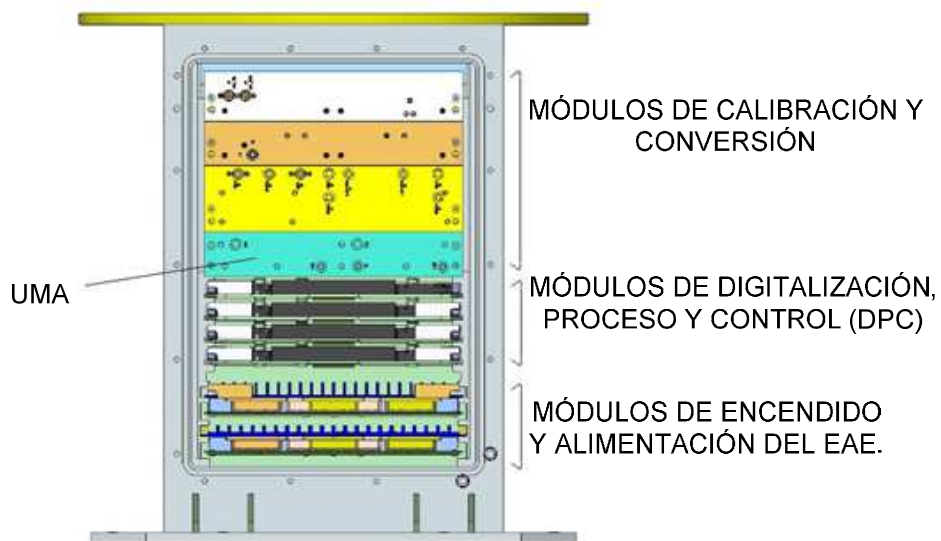


FIGURA 5 : VISTRA FRONTAL DISTRIBUCIÓN DE MÓDULOS DEL EQUIPO DE APUNTAMIENTO EN ELEVACIÓN

En el momento del diseño de la tarjeta TIC, el módulo UMA se presenta ya diseñado con la siguiente forma:

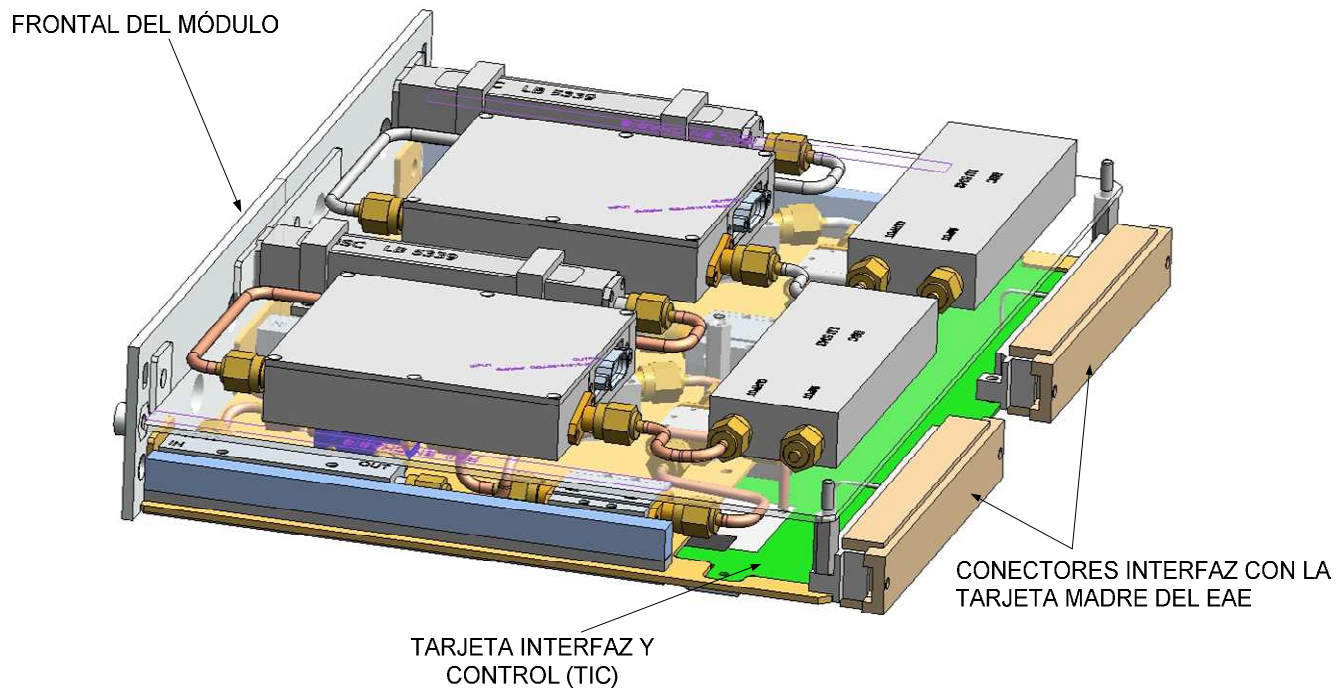


FIGURA 6 : MECÁNICA DEL MÓDULO UNIDAD DE MEZCLA Y ADAPTACIÓN

Como requisito para todos los módulos del EAE viene determinado que deben adaptarse al estándar “Eurocard Doble” o también denominado “Doble Europa”

Como se observa en la figura 6 este módulo presenta una distribución en dos alturas, en la bandeja inferior se ubicará la TIC objeto de este PFC según la siguiente imagen:

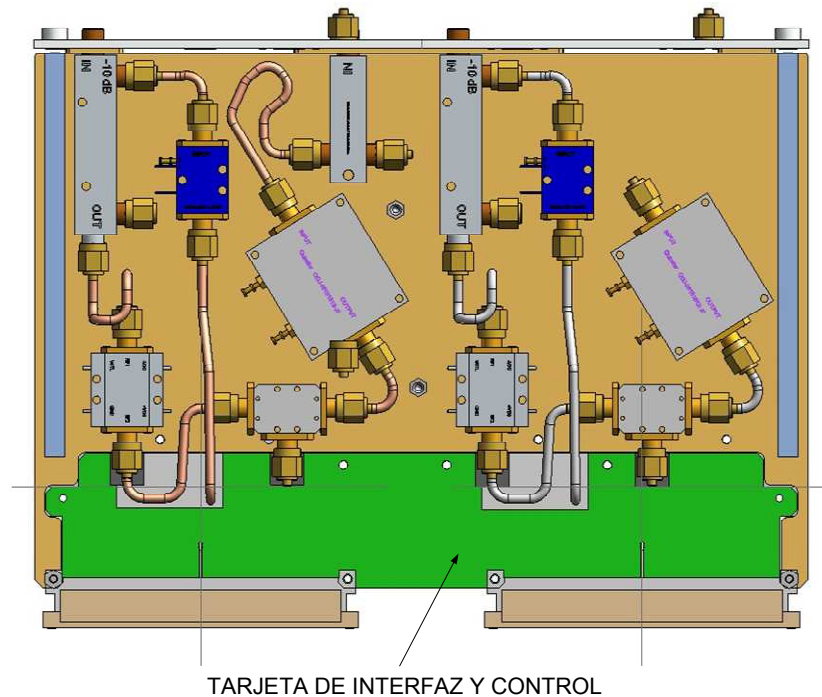


FIGURA 7 : VISTA SUPERIOR DE LA MECÁNICA QUE ALBERGA LA TARJETA INTERFAZ Y CONTROL

El estándar Eurocard es un formato para placas de circuito impreso que puede conectarse en subbastidores estandarizados. Esta estructura consiste en una serie de ranuras en las que se deslizan las tarjetas hasta que se alinean con el frontal. En la parte trasera hay uno o más conectores que se conectan a un “backplane” o tarjeta madre que cierra la trasera del subbastidor. Una Eurocard de tamaño simple es de 100 mm x 160 mm, y una Eurocard de tamaño doble 233,4 x 160 mm.

Eurocard es un sistema mecánico y no define el conector específico a utilizar ni las señales que se asignan a los contactos del conector. Sin embargo, los conectores que se emplean comúnmente con las arquitecturas Eurocard incluyen conectores DIN41612.

Por lo tanto como condición, viene ya establecidas unas cotas que se adaptan al estándar denominado “Eurocard Doble” o “Doble Europa” y cuyas dimensiones a la hora de tener en cuenta para el diseño deben ser de 50x233mm contando el ensamble mecánico y la PCB. La altura máxima de los componentes en la cara de soldaduras viene marcada en 2 mm.

Así mismo, como se observa en las figuras 6 y 7, la TIC contará con conectores que ofrecen una interfaz con el backplane o tarjeta madre del EAE, queda a criterio del diseñador qué tipo de conector DIN41612 emplear. A través de este interfaz con la tarjeta madre se recibirán las

señales de alimentación y control, la distribución de las señales en los conectores queda a criterio del diseñador.

2.2.2. ALIMENTACIÓN DE LA TARJETA.

El EAE distribuye la alimentación a través de una tarjeta madre en la parte trasera del cofre. Según se especifica en el sistema las alimentaciones disponibles que se proporcionan a través de la tarjeta madre son las siguientes:

Alimentación (V)	Tipo
+24	ANALÓGICO
+15	ANALÓGICO
-15	ANALÓGICO
+12	ANALÓGICO
5	DIGITAL
3,3	DIGITAL

TABLA 3 : ALIMENTACIONES DESDE LA TARJETA MADRE

Las características genéricas de las señales de alimentación de entrada a la unidad son:

- Tensión nominal: las marcadas anteriormente.
- Rango de variación: $\pm 1\text{Vdc}$ o $\pm 10\%$ respecto a tensión nominal (el valor más restrictivo).
- Rizado máximo: 50mVpp

Las señales de alimentación de entrada a la TIC ya han sido filtradas. Sin embargo, la unidad debe proporcionar protección adicional contra subidas de tensión en los transitorios de encendido y apagado por medio de los elementos adecuados.

Las características de estos transitorios:

- Duración de los transitorios $< 10\text{ms}$.
- Amplitud de los transitorios $\pm 20\%$ respecto a la tensión nominal

Una de las tareas principales de la Tarjeta Interfaz y Control, como se ha mencionado anteriormente, es distribuir las alimentaciones que necesitan los elementos del módulo UMA. Viene marcado como requisito que dichos elementos necesitan las siguientes:

Alimentación	Tipo
+5V	ANALÓGICO
-5V	ANALÓGICO
+12V	ANALÓGICO
+15V	ANALÓGICO
+8V	ANALÓGICO

TABLA 4 : TENSIONES REGULADAS

La TIC por lo tanto tendrá que ocuparse de tomar las alimentaciones desde la tarjeta madre del equipo, para filtrarlas y regularlas a las tensiones requeridas por los elementos del módulo.

Viene dado como requisito que los elementos que presenta el módulo de Unidad de Mezcla y Adaptación son a grandes rasgos los siguientes:

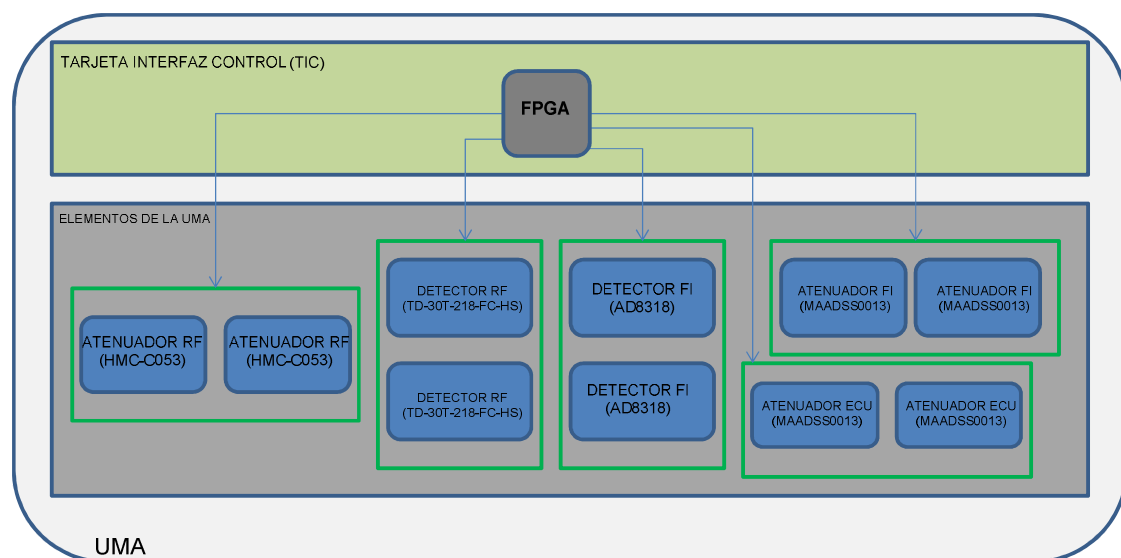


FIGURA 8 : ELEMENTOS CONTENIDOS EN LA UMA

La alimentación de los dispositivos de la figura 8 procederá de la Tarjeta Interfaz y Control (TIC). Para valorar la capacidad que deben tener las pistas así como qué tipo de regulaciones de tensión hay que realizar, se realiza un análisis de consumos totales de los elementos de la UMA. A continuación se detallan los consumos de los elementos que concentran los consumos más significativos:

ELEMENTO		TENSION	CONSUMO	Unidades	CONSUMO TOTAL
PART NAME	FABRICANTE	(V)	(mA)		
HMC-C053	Hittite Microwave	(+5)	3,7	2	7,4
		(-5)	8,8	2	17,6
TD-30T-218-FC-HS	Planar Monolithic Industries	(+12)	100	2	200
AD8318	Analog Devices	(+5)	68	2	136
MAADSS0013	MA-COM Technology Solutions	(+5)	0,25	4	1
REF195	Analog Devices	(+5)	20	2	40
QGJ-06191812-J0	Quinstar Technology	(+12)	500	2	1000
AR2589	Teledyne Microwave Solutions	(+15)	310	4	1240

TABLA 5: CONSUMOS DE LOS ELEMENTOS DE LA TIC

Por último, en la tarjeta de control se deberán incluir los siguientes componentes correspondientes al esquema eléctrico de la figura para realizar el filtrado de alimentación antes de ser distribuidas a otros módulos:

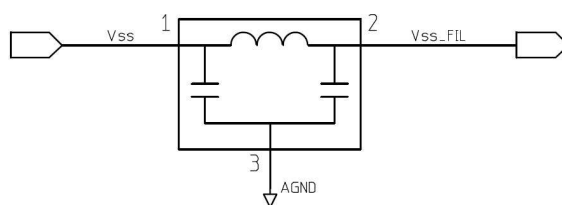


FIGURA 9: ESQUEMA DE FILTRADO

2.2.3. CONDICIONES AMBIENTALES. TEMPERATURA Y HUMEDAD.

Como se ha visto el emplazamiento de los módulos se encuentra dentro del cofre del EAE. Dicho cofre se encontrará expuesto a las condiciones climáticas en la cubierta de un barco. Por lo tanto se deben cumplir una serie de requisitos según las condiciones ambientales.

En cuanto a las temperaturas especificadas la TIC deberá trabajar correctamente en el margen de temperaturas de -25°C a $+50^{\circ}\text{C}$ y así mismo deberá soportar un rango de temperaturas de almacenamiento de -25°C a $+70^{\circ}\text{C}$.

Así mismo, la TIC deberá trabajar correctamente en un ambiente con humedad del 95% en el rango de temperaturas: $+30^{\circ}\text{C}$ a $+50\%$

3. ANALISIS DE ESPECIFICACIONES TÉCNICAS

3.1. ESTUDIO PRELIMINAR DEL DISEÑO

Una vez detallados los requisitos para llevar a cabo el diseño de la Tarjeta de Interfaz y Control objeto de este Proyecto Fin de Carrera, en este apartado se estudiarán dichas especificaciones para encontrar el diseño óptimo que cumpla con las funcionalidades que debe cubrir la TIC.

Para dar soporte al hardware que llevará a término la funcionalidad de la TIC se diseñará una tarjeta de circuito impreso (PCB, printed circuit board) basándose en los requisitos y limitaciones planteados en puntos anteriores.

Se realiza por tanto un diseño hardware preliminar para valorar qué opciones adoptar, y cómo abordar el diseño definitivo.

Para ello será necesario contar con un plano mecánico de la PCB, es decir un plano del perímetro de la tarjeta. Como se indicó en el punto 2.2.1 las limitaciones mecánicas del módulo donde se ubicará la TIC del EAE vienen determinadas ya. Se dispone por tanto del siguiente plano:

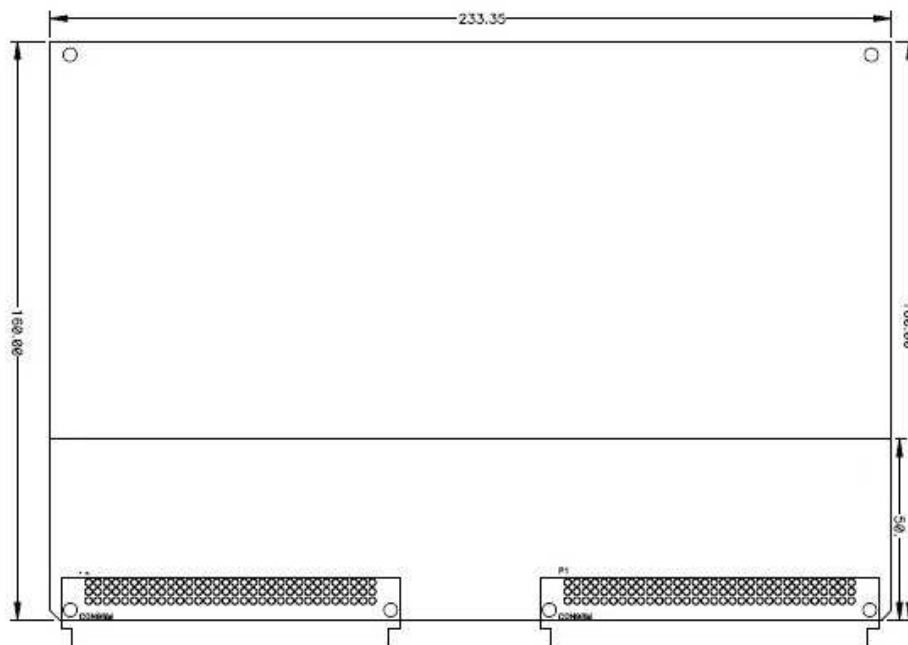


FIGURA 10: PLANO DELINEACIÓN. ESPACIO TARJETA INTERFAZ Y CONTROL

En segundo lugar, es ya ineludible plantear qué componentes son los que deben figurar en la tarjeta y que confeccionarán el comportamiento de la misma.

Para proceder al diseño hardware preliminar, se distinguirá por partes qué acciones básicas se debe desarrollar.

Posteriormente a realizar el estudio de todas las especificaciones y el sistema haya sido dividido en bloques funcionales detallando su arquitectura de entradas/salidas., se abordará el diseño hardware de la PCB.

3.1.1. TRATAMIENTO DE SEÑALES DE CONTROL

De acuerdo a lo mencionado en puntos anteriores, la TIC del EAE debe ocuparse de constituir el funcionamiento de EAE efectuando el control sobre determinadas acciones, que en resumen son las siguientes:

- Control de Frecuencias
- Control de Atenuaciones
- Control de Switches

Para implementar estas funcionalidades habrá que diseñar bloques hardware que cumplan con dicho fin y que irán ubicados en la PCB.

Además, cabe destacar que éste se trata de un diseño de cierta complejidad por lo tanto será necesario disponer de un hardware específico que implemente los diferentes bloques que lleven a cabo todo este control de señales. En definitiva, será necesario disponer de algún dispositivo programable por el usuario con el que implementar el conjunto de funciones que debe llevar a cabo.

Los PLDs (Programmable Logic Device, PLD) constituyen una buena oferta para realizar diseños electrónicos con un buen compromiso coste-prestaciones, con la ventaja de que permiten una implementación en un tiempo de diseño relativamente corto.

Hay varias clases de dispositivos lógicos programables, pero en este Proyecto Fin de Carrera se valoran dos opciones. Se trata de los denominados Field Programmable Gate Array (FPGA) y los Complex Programmable Logic Device (CPLD).

Los circuitos CPLDs y FPGAs suministran una solución reconfigurable y eficiente para implementar todo este diseño de hardware digital que se necesita en este Proyecto Fin de Carrera.

Las principales diferencias entre CPLDs y FPGAs son relacionados a su arquitectura. Un CPLD tiene una estructura un tanto restrictiva, lo que supone menor flexibilidad. Las arquitecturas FPGAs, por otro lado, están dominadas por la interconexión lo que las hace más flexibles.

En la práctica, la elección entre FPGAs o CPLDs a menudo consiste en que el primero es generalmente más denso, contienen más funciones embebidas tales como sumadores, multiplicadores, bloques de memoria, registros, etc.... Por otro lado, los CPLDs contienen flash interna para almacenar su configuración mientras que las FPGAs son volátiles y utilizan una

memoria externa PROM para almacenar la configuración. Además de esto, otros criterios básicos son el volumen del diseño, la frecuencia necesaria, el coste, la tipología del diseño según la necesidad de elementos específicos, etc...

Habitualmente las FPGAs son de mayor capacidad y resultan más rentables al diseñador. Ésta definitivamente es la opción elegida para implementar el diseño de la lógica programable

El diseño del circuito de la FPGA se realizará empleando un lenguaje de descripción de hardware (HDL), por lo que implementar los bloques de hardware digital necesarios para el control de señales requerido, consiste en capturar el diseño hardware en VHDL o Verilog.

En este punto del PFC cabe descartar que Xilinx y Altera son los líderes actuales del mercado y rivales de la industria. Juntos controlan más del 80% del mercado, Xilinx sola representa más del 50%.

3.1.1.1. COMUNICACIONES DE LA TIC DEL EAE

Según se especificó en el apartado 2.1, para ciertas comunicaciones entre la TP y la TIC del EAE se dispone de un puerto serie RS-232. Este puerto permitirá intercambiar información entre TP y TIC, mediante tres hilos físicos entre ambas tarjetas: TX, RX y GND.

En las comunicaciones serie RS-232 los valores de los voltajes para representar los 1's y 0's lógicos difieren de los TTL, donde 5V para el '1' y 0V para el '0'. Para estas comunicaciones la línea en reposo está a nivel lógico alto (-15V), como se ve en la siguiente figura:

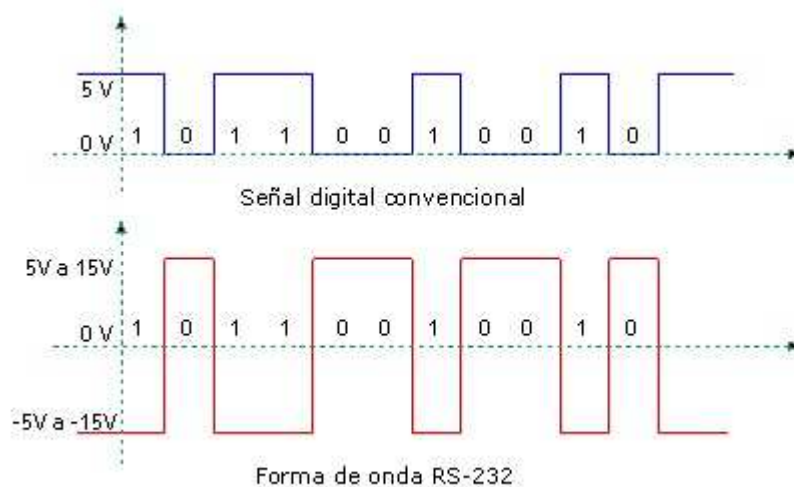


FIGURA 11 : FORMAS DE ONDA RS-232

Debido a que los niveles de tensión del estándar RS-232 no son compatibles con los niveles TTL habituales, se requiere un transceiver que convierta estas tensiones de unos niveles a otros y viceversa.

3.1.1.2. CONTROL DE FRECUENCIAS

De acuerdo a los requisitos especificados, se dispone de un bus de 16 líneas TTL desde la Tarjeta Procesadora que proporcionarán la información apropiada para llevar a cabo el Control de Frecuencias.

Debido a la resolución del sistema, se identifica que para cubrir todo el rango de frecuencias son necesarios 20 bits. Por lo tanto la Tarjeta Procesadora debe ocuparse de entregar estos 20 bits de información utilizando las 16 líneas físicas disponibles que existen de interfaz a la TIC.

Además se ha marcado como requisito que la TP tiene que pasar un valor de Desviación que se empleará en calibración.

Debido a que no hay líneas suficientes entre la TP y la TIC se adopta como solución realizar este procedimiento mandando la información a la TIC en dos tandas de la siguiente manera:

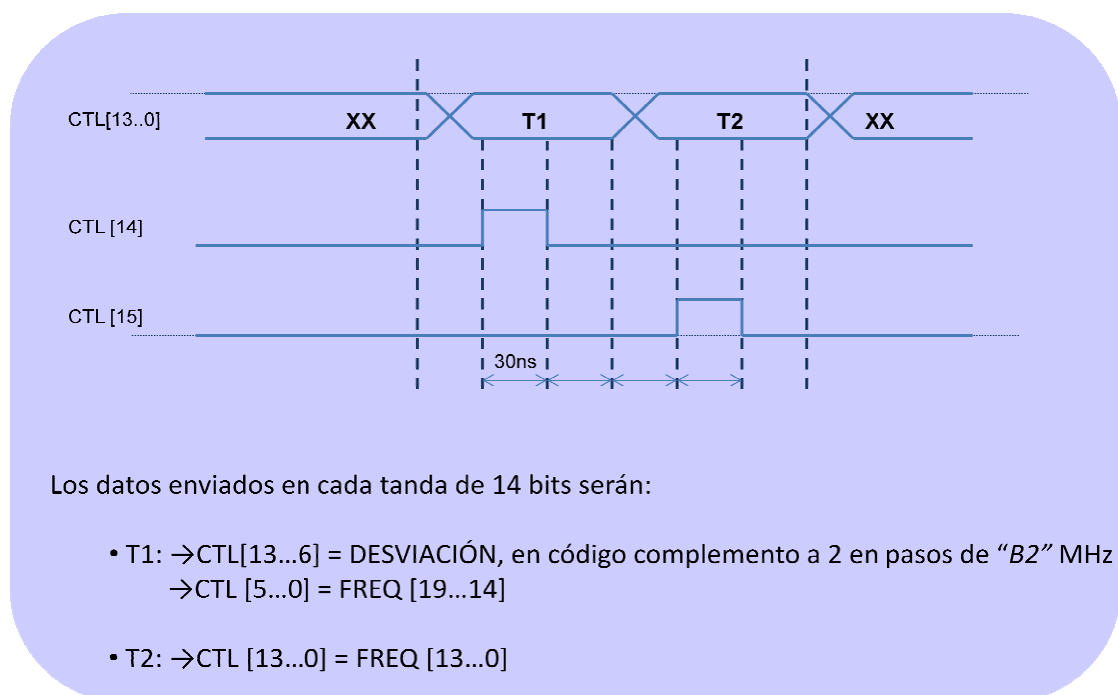


FIGURA 12 : CAPTURA DE BUS DE FRECUENCIA

Una vez capturado el dato por la TIC los buses de líneas con la información del control de frecuencia deben quedar con la siguiente estructura:

BUS FRECUENCIA									
BIT19	BIT18	BIT17	BIT16	BIT15	BIT3	BIT2	BIT1	BIT0
FREQ19	FREQ18	FREQ17	FREQ16	FREQ15	FREQ3	FREQ2	FREQ1	FREQ0

TABLA 6 : DATO FRECUENCIA

BUS DESVIACION (código Complemento-a-2)							
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
DESV8	DESV7	DESV6	DESV5	DESV4	DESV3	DESV1	DESV0

TABLA 7 : DATO DESVIACIÓN

Teniendo en cuenta la resolución de B1 kHz y el rango de F1 a F2 GHz, se prepara la siguiente tabla que detalla para cada frecuencia qué valores pasa la TP, los cuales compondrán el dato que la TIC debe capturar y posteriormente adaptar.

FRECUENCIA DE TRABAJO (kHz)	CÓDIGO FRECUENCIA (señales TTL adaptadas en la TIC)
0,00	0000 0000 0000 0000 0000
1·B1	0000 0000 0000 0000 0001
2·B1	0000 0000 0000 0000 0010
3·B1	0000 0000 0000 0000 0011
4·B1	0000 0000 0000 0000 0100
.....
.....
$F1 = n_1 \cdot B1$	XXXX XXXX XXXX XXXX XXXX
.....
$F2 = n_2 \cdot B1$	XXXX XXXX XXXX XXXX XXXX
.....

TABLA 8 : CÓDIGO FRECUENCIA

Como se explicó en las especificaciones del apartado 2 los elementos a los que programar la frecuencia de trabajo son:

ELEMENTO RF	SEÑALES TTL para la FRECUENCIA
FILTRO SINTONIZABLE YIG	FREQ_YIG[11...0]
SINTETIZADOR DE SINTONÍA	FREQ_SINT[19...0]
SINTETIZADOR DE CALIBRACIÓN	FREQ_CAL[14...0]

TABLA 9 : ELEMENTOS RF CON FRECUENCIA SINTONIZABLE

Por lo tanto este bloque funcional se implementará como código de la FPGA, y no será necesario emplear ningún elemento hardware para este control.

De estas mismas unidades llegaran a la TIC del EAE, a modo de entradas a la FPGA, las siguientes señales de comprobación:

UNIDAD	SEÑALES DE COMPROBACION
UFA	BITE_UFA_0 y BITE_UFA_1
UOL	BITE_SINT
UC	BITE_CAL

TABLA 10 : SEÑALES DE AUTOCOMPROBACIÓN DEL EAE

3.1.1.3. CONTROL DE UMBRALES

Viene determinado que dado que la detección de nivel en banda tan ancha no es plana, se necesita conmutar umbrales en función de la frecuencia para mantener en lo posible el mismo nivel de disparo. Se necesitara por lo tanto incluir algún dispositivo en la Tarjeta Interfaz y Control de EAE que se ocupe de llevar a cabo esta tarea.

La idea consiste en generar diferentes tensiones umbrales y poder conmutar entre ellas según la frecuencia de sintonía marcada por la TP.

Además como requisito se solicita que las bandas de frecuencia puedan sufrir variaciones en función de las medidas.

En resumen, cada frecuencia requiere un nivel de tensión, por lo tanto en el caso hipotético que trabajásemos solo con una frecuencia se determinaría colocar en este diseño un potenciómetro y realizar los ajustes adecuados hasta encontrar el nivel de umbral necesario.

Como en realidad se va a contar con una variedad de frecuencias, se necesitará emplear una resistencia programable con la que podemos configurar para cada caso el valor umbral. Se necesitará por lo tanto encontrar un dispositivo que cumpla estas especificaciones e incluirlo en la PCB.

3.1.1.4. CONTROL DE ATENUACIÓN

Las señales de control para los atenuadores ubicados en la UMA (Unidad de Mezcla y Adaptación) serán tratadas en la TIC del EAE desde la lógica programable. Con la información procedente de la Tarjeta Procesadora, desde el código implementado en la FPGA se generaran valores digitales que se correspondan a cada uno de los valores de los atenuadores.

Como se mencionó en el punto 2.1.3, la atenuación se realizará en tres etapas en diferentes puntos de la cadena de RF.

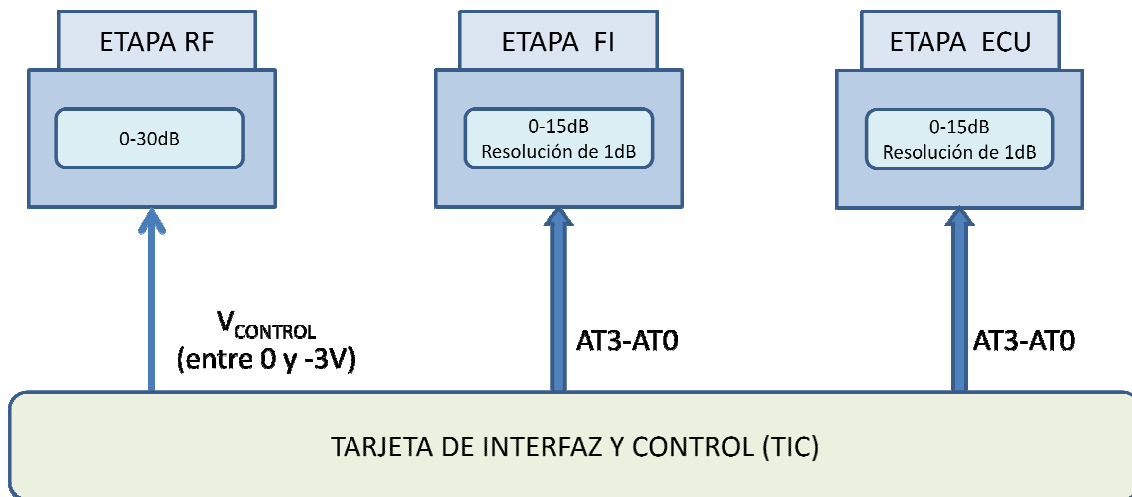


FIGURA 13 : ATENUACIÓN EN TRES ETAPAS

Para el caso del atenuador *MAADSS0013* del fabricante *Tyco Electronics*, éste especifica la siguiente tabla de verdad:

VC1	VC2	VC4	VC8	Attenuation (dB)
0	0	0	0	Reference IL
1	0	0	0	1
0	1	0	0	2
0	0	1	0	4
0	0	0	1	8
1	1	1	1	10
Note: '0'=0V, '1'=+2.8 to 5V				

TABLA 11 : TABLA DE VERDAD MAADSS0013

Desde la FPGA será fácil implementar un proceso que se ocupe de generar el código digital adecuado según el comando en dB de la Tarjeta Procesadora. Por lo tanto no será necesario diseñar ningún bloque hardware en la PCB de la TIC del EAE para esta tarea.

Pero para el caso del atenuador *HMC-C053* de *Hittite Microwave Corporation*, tal y como se indicó en el apartado 2.1.3 se dispone de 30 dB de atenuación, que son controlados con una línea de tensión de control entre 0V y -3V. El fabricante indica mediante la siguiente figura una relación de atenuación en función del voltaje:

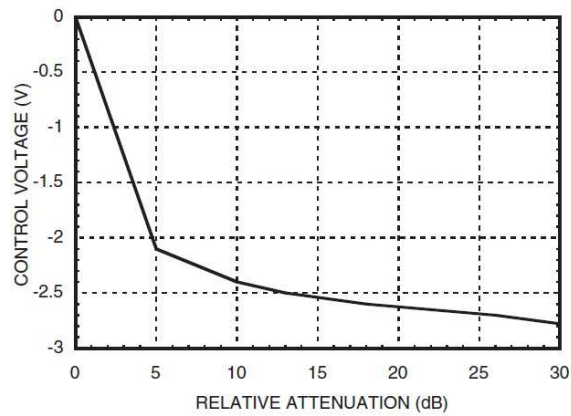


FIGURA 14 : GRÁFICA FUNCIONAMIENTO DE HMC-C053

Para implementar esta señal a partir del código generado desde la lógica programable, se deberá tratar el valor indicado por la TP y proporcionar una señal analógica entre 0 y -3V. Habrá que llevar a cabo la implementación de un bloque hardware que se ocupe de esta tarea, que linealmente genere este rango de tensiones. Será la TP la que decida qué valor de tensión programar.

Finalmente desde ciertos detectores de la UMA llegaran a la TIC del EAE, a modo de entradas a la FPGA, las siguientes señales de comprobación:

SEÑALES DE COMPROBACION
BITE_CANAL1 Y BITE CANAL2
OVERLOAD1 Y OVERLOAD2

TABLA 12 : SEÑALES DE AUTOCOMPROBACIÓN DE LA UMA

3.1.1.5. CONTROL DE SWITCHES

De acuerdo a lo mencionado en el apartado 2.1.2 existen dos modos de funcionamiento que actúan en el direccionamiento de las señales de RF como se observa en la siguiente figura:

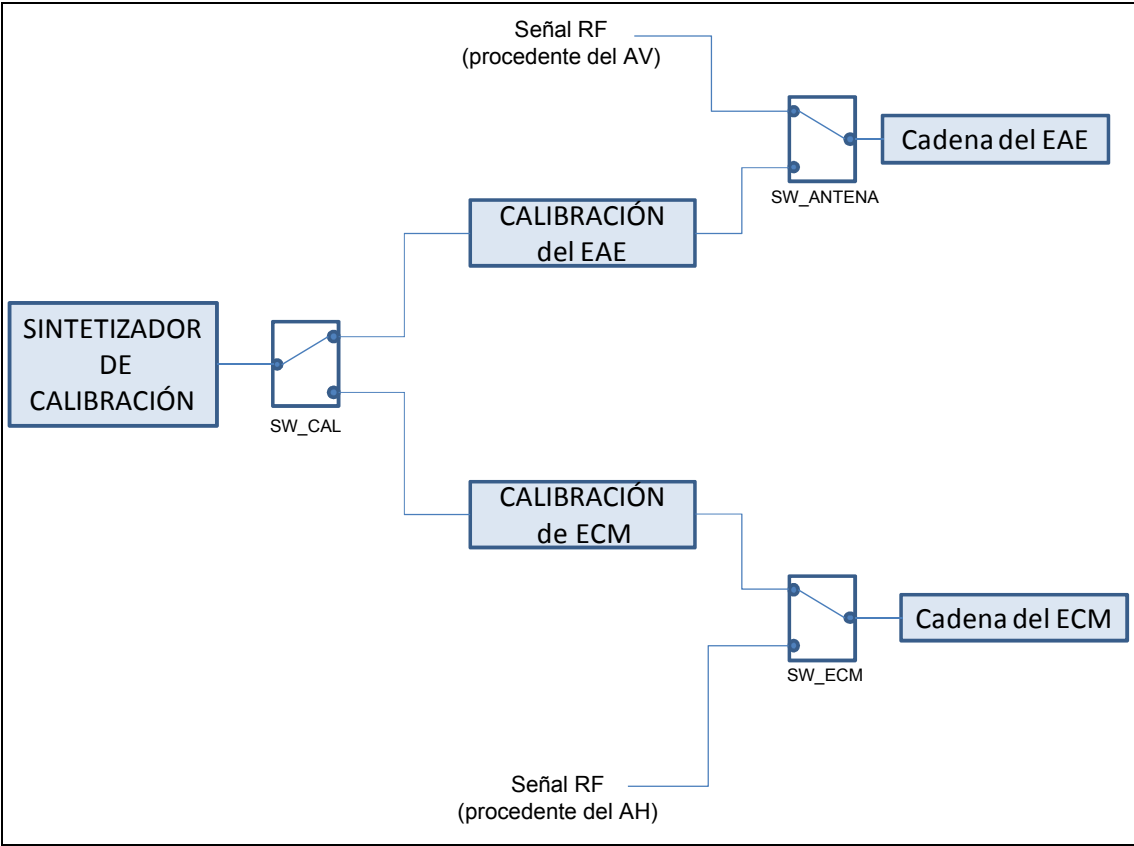


FIGURA 15 : DIRECCIONAMIENTO SEÑALES DE RF. CONTROL DE SWITCHES

Para llevar a cabo este control, será necesario prestar atención a qué modo de funcionamiento desea programar la TP. Para ello, se especifica las siguientes posiciones de los switches:

SWITCH_ANTENNA[1...0]		
MODO	BIT 1	BIT 0
MODO ANTENA	0	1
MODO CALIBRACION	1	0

SWITCH_ECM[1...0]		
MODO	BIT 1	BIT 0
MODO ANTENA	0	1
MODO CALIBRACION	1	0

SWITCH_CAL[1...0]		
MODO	BIT 1	BIT 0
CALIBRACION EAE	0	1
CALIBRACION ECM	1	0

SWITCH_FILTRO[1...0]		
MODO	BIT 1	BIT 0
CON FILTRO	0	1
SIN FILTRO	1	0

Para ello, la TP escribirá mediante el puerto serie en el registro oportuno el modo de funcionamiento que desea programar y la TIC será capaz de llevar el manejo de las señales de cada switch según el modo programado.

Además dentro de este mismo bloque funcional se incluirá la señal de “BLANKING” mencionada anteriormente. Recuérdese que se trata de una señal procedente desde el cofre del EAE y que es indicativo de que se están realizando contramedidas. La parte que le toca a la TIC es ocuparse de accionar los switches del modo siguiente:

Blanking			
VALOR	SW_ECM	SW_ANTENA	SW_CAL
0	MODO ANTENA	MODO ANTENA	NO APLICA
1	MODO CALIBRACION	MODO CALIBRACION	NO APLICA

En definitiva, esta parte del control que debe ejercer la TIC se implementará dentro de la FPGA.

3.1.1.6. FILTRADO DE SEÑALES DE CONTROL

Por otro lado, como se ha detallado en los requisitos, la interfaz de señales a la TIC del EAE es principalmente TTL es decir entradas y salidas de 5V. La TIC del EAE sin embargo contará con una FPGA cuyas entradas y salidas son de 3,3V (LVTTTL). Resulta imprescindible pasar las señales por alguna etapa que adapte estos niveles para la FPGA y además que sirvan de protección a la misma por posibles daños en picos de tensión. Se necesitará un bloque HW como el siguiente:

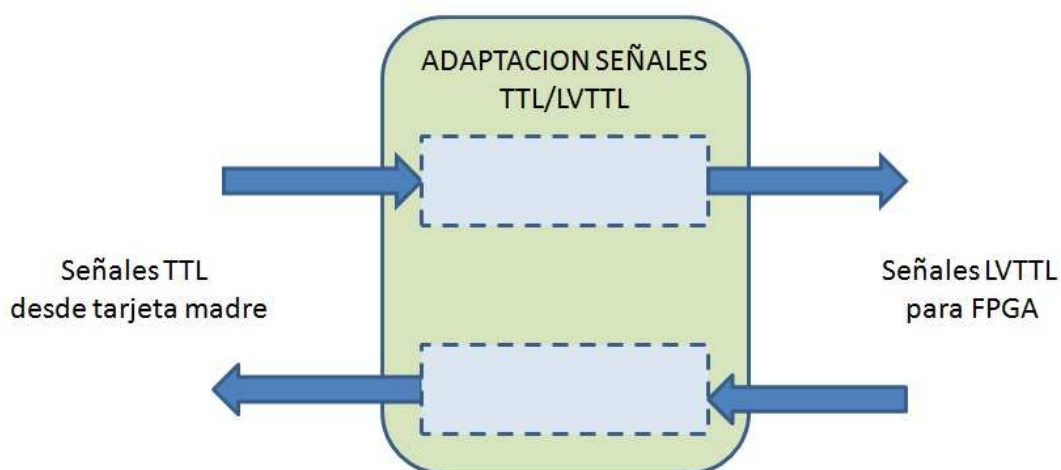


FIGURA 16 : ADAPTACIÓN DE SEÑALES TTL-LVTTTL

3.1.2. FILTRADO DE ALIMENTACIONES

Para eliminar posibles ruidos desde el cofre del EAE y desde los propios elementos de la UMA, se decide que es necesario filtrar las alimentaciones digitales de 5VDIG y 3,3VDIG que entran en la tarjeta. Por lo tanto se tendrá que implementar un bloque HW del tipo:

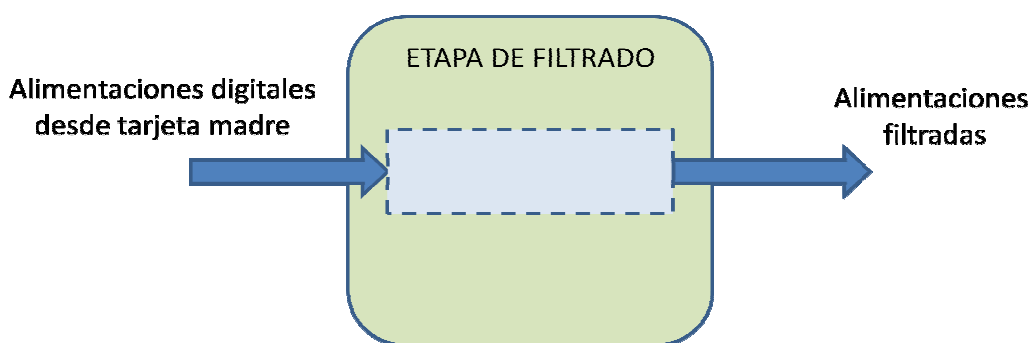


FIGURA 17 : ETAPA DE FILTRADO SEÑALES DIGITALES

3.1.3. ADAPTACIÓN DE ALIMENTACIONES

Como se ha destacado anteriormente, la alimentación procede de la tarjeta madre del EAE y una de las tareas de las que se ocupa la TIC del EAE es de suministrar las alimentaciones requeridas por los módulos empleados en la UMA (Unidad de Mezcla y Adaptación). Para ello se implementará en la PCB un circuito de regulación de tensión.

Son necesarias las siguientes alimentaciones en la unidad:

- $\pm 5V$
- +12V
- +15V
- +8V

Teniendo en cuenta las alimentaciones que recibirá la tarjeta desde la tarjeta madre del equipo, será oportuno por lo tanto diseñar en la PCB un bloque HW como el siguiente:

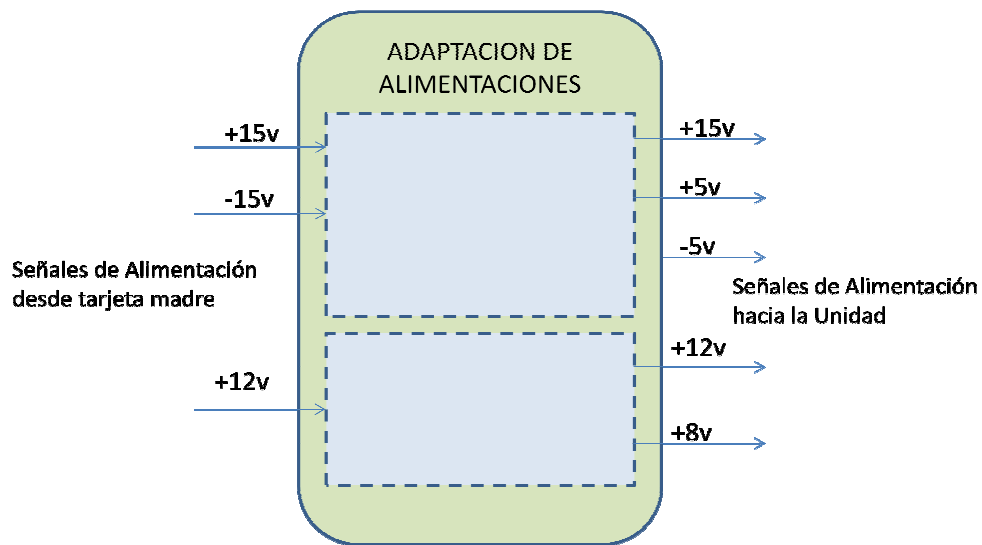


FIGURA 18: ADAPTACIÓN DE ALIMENTACIONES

3.2. SOLUCIÓN ADOPTADA.

El diseño se abordará posteriormente a tener analizadas las especificaciones y el sistema haya sido dividido en bloques generales detallando su arquitectura de entradas/salidas. Mientras la placa de circuito impreso esté siendo fabricada se desarrollará el código de descripción hardware, se simulará y se depurará el diseño ya que esto no afecta a la placa sino a los circuitos programables incluidos en ella.

Pero antes de todo ello, se debe comenzar por presentar un diagrama genérico a grandes rasgos de los bloques funcionales de la tarjeta.

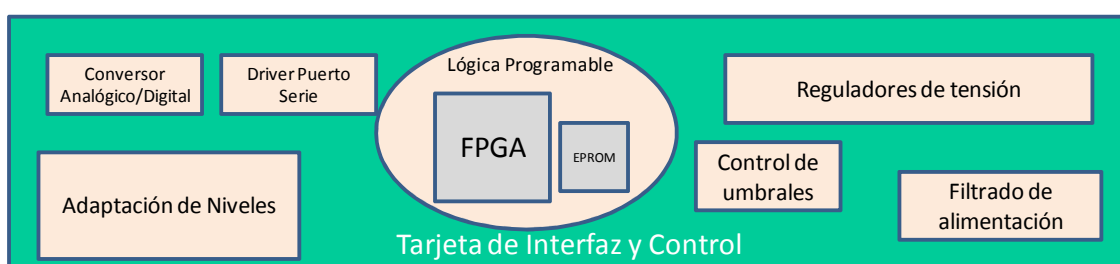


FIGURA 19 : BLOQUES FUNCIONALES HARDWARE DE LA TIC

A lo largo de este apartado, y con este primer diagrama, se puede ya empezar a desarrollar la solución más conveniente eligiendo los materiales óptimos para el diseño tomando como base principal el reducido espacio con el que se cuenta.

Como norma principal a la hora de acometer un nuevo diseño, se deberán elegir componentes que cumplan las condiciones ambientales aplicables al diseño. Por otro lado, el método para elegir componentes será la búsqueda de los mismos en la Lista de Componentes Preferenciales de INDRA SISTEMAS. De estos componentes se almacena información como el detalle de unidades en stock o documentación de fabricante. En caso de no encontrar el componente se acudirá a buscar nuevos componentes.

3.2.1. INTERFACES DE E/S EN LA TIC.

Una vez realizado el estudio preliminar del diseño de la TIC del EAE, objeto de este Proyecto Fin de Carrera, y una vez desglosadas las funcionalidades por bloques, ya se puede realizar una lista de señales Entrada/Salida de la tarjeta.

La siguiente figura muestra un diagrama de interconexión entre tarjetas:

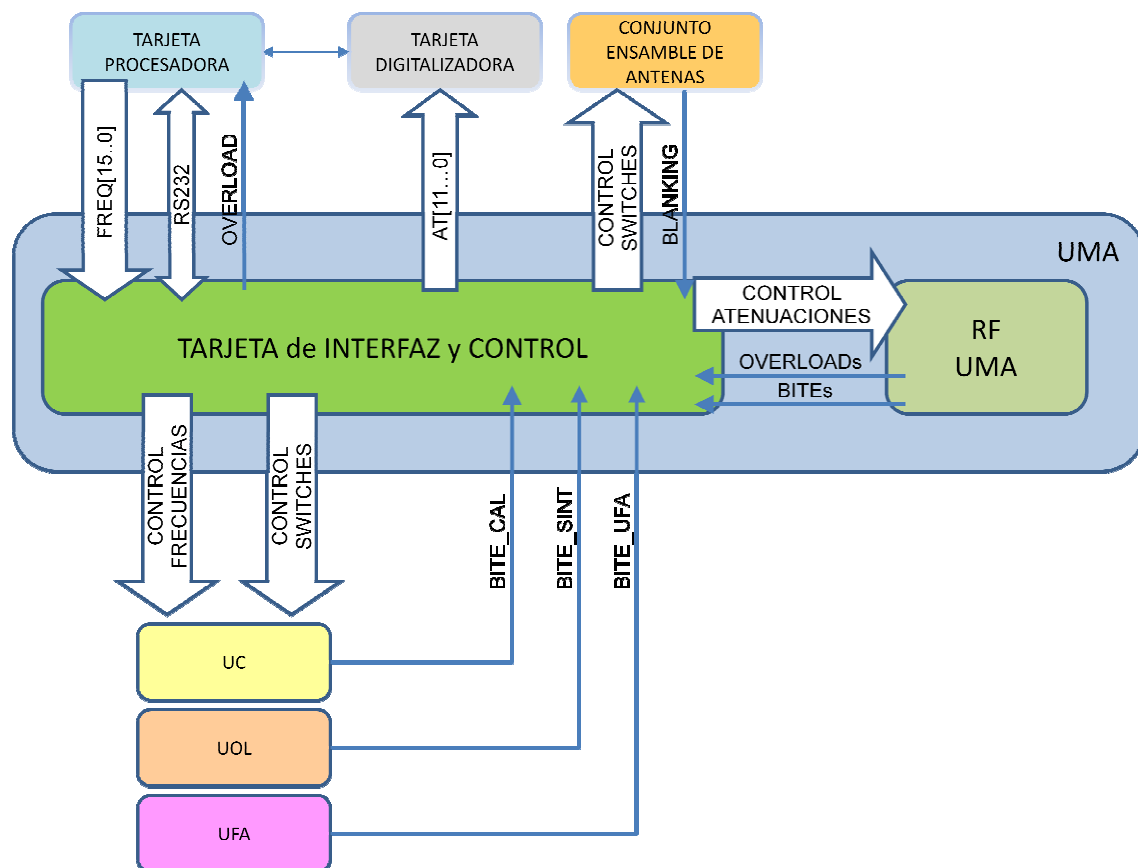


FIGURA 20 : DIAGRAMA DE INTERCONEXIÓN DE LA TIC

La siguiente tabla detalla las líneas de señales de control:

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
FREQ15	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ14	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ13	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ12	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ11	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ10	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ9	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ8	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ7	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ6	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ5	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ4	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ3	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ2	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ1	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
FREQ0	TTL	TP		TIC	FPGA	CONECTORES PLACA MADRE
RX	Estándar RS-232	TP		TIC	FPGA	CONECTORES PLACA MADRE
TX	Estándar RS-232	TIC	FPGA	TP		CONECTORES PLACA MADRE

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
AT11	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT10	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT9	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT8	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT7	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT6	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT5	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT4	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT3	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT2	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT1	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT0	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
BLANKING	TTL	TIC	FPGA	TD	FPGA	CONECTORES PLACA MADRE
AT_FI1_0	TTL	TIC	FPGA	UMA	ATENUADOR CH1 FI	CONECTOR MODULOS UMA
AT_FI1_1	TTL	TIC	FPGA	UMA	ATENUADOR CH1 FI	CONECTOR MODULOS UMA
AT_FI1_2	TTL	TIC	FPGA	UMA	ATENUADOR CH1 FI	CONECTOR MODULOS UMA
AT_FI1_3	TTL	TIC	FPGA	UMA	ATENUADOR CH1 FI	CONECTOR MODULOS UMA
AT_ECU1_0	TTL	TIC	FPGA	UMA	ATENUADOR CH1 ECU	CONECTOR MODULOS UMA

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
AT_ECU1_1	TTL	TIC	FPGA	UMA	ATENUADOR CH1 ECU	CONECTOR MODULOS UMA
AT_ECU1_2	TTL	TIC	FPGA	UMA	ATENUADOR CH1 ECU	CONECTOR MODULOS UMA
AT_ECU1_3	TTL	TIC	FPGA	UMA	ATENUADOR CH1 ECU	CONECTOR MODULOS UMA
VCTL_1	ANALOG	TIC	FPGA	UMA	ATENUADOR CH1 RF	TIRA DE PINES
UMBRAL_FI_1	ANALOG	TIC	POTENCIOMETRO	UMA	DETECTOR	TIRA DE PINES
BITE_CANAL1	TTL	UMA	DETECTOR	UMA	FPGA	CONECTOR MODULOS UMA
OVERLOAD1	TTL	UMA	DETECTOR	UMA	FPGA	CONECTOR MODULOS UMA
AT_FI2_0	TTL	TIC	FPGA	UMA	ATENUADOR CH2 FI	CONECTOR MODULOS UMA
AT_FI2_1	TTL	TIC	FPGA	UMA	ATENUADOR CH2 FI	CONECTOR MODULOS UMA
AT_FI2_2	TTL	TIC	FPGA	UMA	ATENUADOR CH2 FI	CONECTOR MODULOS UMA
AT_FI2_3	TTL	TIC	FPGA	UMA	ATENUADOR CH2 FI	CONECTOR MODULOS UMA
AT_ECU2_0	TTL	TIC	FPGA	UMA	ATENUADOR CH2 ECU	CONECTOR MODULOS UMA
AT_ECU2_1	TTL	TIC	FPGA	UMA	ATENUADOR CH2 ECU	CONECTOR MODULOS UMA
AT_ECU2_2	TTL	TIC	FPGA	UMA	ATENUADOR CH2 ECU	CONECTOR MODULOS UMA
AT_ECU2_3	TTL	TIC	FPGA	UMA	ATENUADOR CH2 ECU	CONECTOR MODULOS UMA
VCTL_2	ANALOG	TIC	FPGA	UMA	ATENUADOR CH2 RF	TIRA DE PINES
UMBRAL_FI_2	ANALOG	TIC	POTENCIOMETRO	UMA	DETECTOR	TIRA DE PINES
BITE_CANAL2	TTL	UMA	DETECTOR	TIC	FPGA	CONECTOR MODULOS UMA

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
OVERLOAD2	TTL	UMA	DETECTOR	TIC	FPGA	CONECTOR MODULOS UMA
FREQ_SINT20	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT19	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT18	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT17	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT16	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT15	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT14	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT13	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT12	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT11	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT10	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT9	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT8	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT7	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT6	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT5	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT4	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
FREQ_SINT3	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT2	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT1	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
FREQ_SINT0	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
STROBE_SINT	TTL	TIC	FPGA	UOL	SINTETIZADOR DE SINTONIA	CONECTORES PLACA MADRE
BITE_SINT	TTL	UOL	DETECTOR	TIC	FPGA	CONECTORES PLACA MADRE
FREQ_CAL14	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL13	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL12	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL11	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL10	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL9	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL8	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL7	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL6	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL5	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL4	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL3	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
FREQ_CAL2	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL1	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
FREQ_CAL0	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
STROBE_CAL	TTL	TIC	FPGA	UC	SINTETIZADOR DE CALIBRACION	CONECTORES PLACA MADRE
BITE_CAL	TTL	UC	DETECTOR	TIC	FPGA	CONECTORES PLACA MADRE
SWITCH_CAL_0	TTL	TIC	FPGA	UC	SWITCH	CONECTORES PLACA MADRE
SWITCH_CAL_1	TTL	TIC	FPGA	UC	SWITCH	CONECTORES PLACA MADRE
FREQ_YIG11	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG10	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG9	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG8	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG7	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG6	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG5	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG4	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG3	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG2	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
FREQ_YIG1	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
FREQ_YIG0	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
STROBE_YIG	TTL	TIC	FPGA	UFA	FILTRO YIG	CONECTORES PLACA MADRE
BITE_UFA	TTL	UFA	DETECTOR	TIC	FPGA	CONECTORES PLACA MADRE
SWITCH_FILTRO_0	TTL	TIC	FPGA	UFA	SWITCH 1	CONECTORES PLACA MADRE
SWITCH_FILTRO_1	TTL	TIC	FPGA	UFA	SWITCH 1	CONECTORES PLACA MADRE
SWITCH_ECM_0	TTL	TIC	FPGA	UFA	SWITCH 2	CONECTORES PLACA MADRE
SWITCH_ECM_1	TTL	TIC	FPGA	UFA	SWITCH 2	CONECTORES PLACA MADRE
JTAG_TDI	TTL	PC		TIC	FPGA	TIRA DE PINES
JTAG_TMS	TTL	PC		TIC	FPGA	TIRA DE PINES
JTAG_TCK	TTL	PC		TIC	FPGA	TIRA DE PINES
JTAG_TDO	TTL	TIC	FPGA	PC		TIRA DE PINES
SWITCH_ANTENA_0	TTL	TIC	FPGA	ANTENAS COFRE EAE		CONECTORES PLACA MADRE
SWITCH_ANTENA_1	TTL	TIC	FPGA	ANTENAS COFRE EAE		CONECTORES PLACA MADRE

TABLA 13 : DESCRIPCIÓN DE LÍNEAS DE SEÑALES DE CONTROL

La siguiente tabla detalla las líneas de señales de alimentación:

NOMBRE		ORIGEN		DESTINO		INTERFAZ
NOMBRE SEÑAL	TIPO	TARJETA ORIGEN	ELEMENTO	TARJETA DESTINO	ELEMENTO	
+15V	ANALOG	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
-15V	ANALOG	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
+12V	ANALOG	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
+5VD	TTL	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
+3.3VD	LVTTTL	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
AGND	ANALOG	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE
DGND	TTL	TARJETA MADRE EAE		TIC		CONECTORES PLACA MADRE

TABLA 14: DESCRIPCION SEÑALES DE LINEAS DE ALIMENTACION

3.2.2. BLOQUES HARDWARE.

3.2.2.1. DISPOSITIVO DE LÓGICA PROGRAMABLE

Como se ha mencionado en puntos anteriores, es necesario un dispositivo programable con el que abordar el diseño de las funciones lógicas.

Por filosofía de empresa se intentará reutilizar en la medida de lo posible tanto material como conocimientos y experiencias anteriores para abordar los diseños. En anteriores diseños de tarjetas interfaces entre módulos similares a la que hoy es objeto de desarrollo, se han empleado FPGAs sencillas y de manejo simple, ya que en estos casos el diseño no requiere de voluminosa lógica programable. Además volver a emplear el mismo dispositivo es una buena práctica para optimizar el tiempo de estudio de la configuración de la FPGA y además se pueden aprovechar todas las librerías con las que se cuenta ya en el entorno de trabajo del departamento en cuestión.

Utilizaremos la familia SPARTAN IIE, que típicamente son utilizadas en aplicaciones de alto volumen y donde la versatilidad de una solución rápida se considera una característica favorable.

La familia Spartan –IIE ofrece a los usuarios un alto rendimiento, abundantes recursos lógicos, y un rico conjunto de características, todo ellos a un precio excepcionalmente bajo. Además ofrece densidades que van desde 50.000 a 600.000 puertas del sistema. Los dispositivos ofrecen velocidades de reloj más allá de 200MHz. Todo ello más que suficiente para el diseño que se quiere llevar a cabo en este Proyecto Fin de Carrera.

En resumen, la arquitectura de esta FPGA contiene cinco elementos principales (ver figura 21):

- Bloque de entrada-salida (IOB o Input-Output Block): estos bloques proporcionan la interfaz entre los pines del integrado y la lógica interna. Cada IOB controla un pad externo.
- Bloque lógico configurable (CLB o Configurable Logic Block): Estos son los bloques básicos que se utilizarán en la implementación de un circuito digital.
- Bloque de distribución y compensación de reloj (DLL o Delay Locked Loop): Estos bloques controlan los dominios de reloj dentro del integrado y compensan por retardos que pueda haber entre el reloj externo y el interno.

- Bloque de memoria (BLOCK RAM): Estos bloques son memorias dedicadas integradas dentro de la lógica programable.
- Estructura de interconexión: Es una estructura versátil y multi-nivel de interconexión entre los otros componentes de la FPGA.

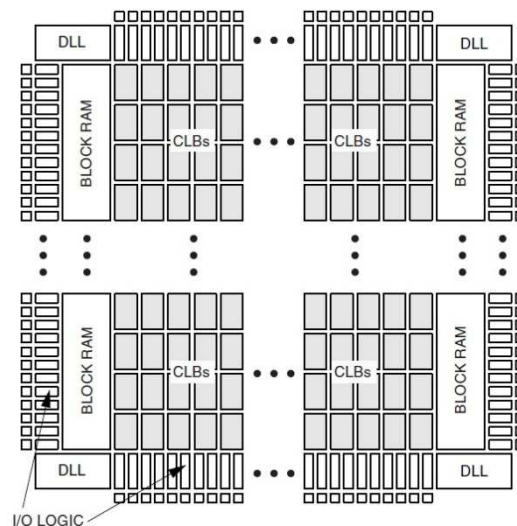


FIGURA 21 : DIAGRAMA DE BLOQUES DE LA FAMILIA SPARTAN-IIIE

Para reducir la potencia consumida por las FPGAs el núcleo se alimenta con una tensión menor que los bloques I/O. cuyas patillas funcionan a 3.3 y cuyo core con 1.8V. Lo que supone que en este diseño se deben generar dichas alimentaciones bien sea por tarjeta madre del equipo o bien desde la propia tarjeta.

Las FPGAs Spartan-IIIE se personalizan al cargar los datos de configuración en las células de memoria estática interna. Con este enfoque son posibles ciclos de reprogramación ilimitados. Los valores almacenados en estas células determinan las funciones lógicas y las interconexiones implementadas en la FPGA.

Xilinx indica que el método más común de programar las FPGAs es mediante la utilización de PROMs conectados a la cadena de la FPGA. Los datos de configuración pueden ser leídos desde una memoria PROM externa (*master serial mode*) o escritos en la FPGA (*slave serial, slave parallel or Boundary Scan modes*). Se elegirá el primer método de configuración para este Proyecto Fin de Carrera.

Xilinx aporta un listado completo para cada elemento de la familia Spartan IIE cuál es la memoria compatible para la FPGA en cuestión. La familia de memorias que determina compatible en este caso es la serie XC18V00, en concreto:

Device	XC18V00 Solution
XC2S300E	XC18V02

TABLA 15 : RELACIÓN FPGA-PROM

Por lo tanto la memoria que se empleará para este Proyecto Fin de Carrera será el dispositivo XC18V02.

La configuración es el proceso por el cual el bitstream de un diseño, generado mediante el software de Xilinx, es cargado dentro de la memoria de configuración interna de la FPGA Spartan IIE.

Para el caso de la SPARTANIIE estos son los modos de configuración:

Configuration Mode	Preconfiguration Pull-ups	M0	M1	M2	CCLK Direccion	Data Width	Serial D _{OUT}
Master Serial mode	No	0	0	0	Out	1	Yes
	Yes	0	0	1			
Slave Parallel mode	No	0	1	0	In	8	No
	Yes	0	1	1			
Boundary-Scan mode	No	1	0	0	N/A	1	No
	Yes	1	0	1			
Slave Serial mode	No	1	1	0	In	1	Yes
	Yes	1	1	1			

TABLA 16 : MODOS DE CONFIGURACIÓN

La configuración a través del puerto Boundary-Scan está siempre disponible, independiente de la selección de modo. Al seleccionar el modo Boundary-Scan, simplemente se apagan los otros modos. Los tres pines de modo tienen resistencias internas de pull-up si por defecto se dejan desconectados.

Hay dos tipos de pines que se utilizan en la configuración de la Spartan-IIE: los pines dedicados y los pines de propósito general.

Los pines dedicados incluyen los pines de modo (M2, M1, M0), el pin de reloj de configuración (CCLK), el pin de PROGRAM, el pin de DONE y los pines de Boundary Scan (TDI, TDO, TMS, TCK).

Además configurar la Spartan-IIE en el modo Master Serial supone establecer los pines de modo (M0, M1, M2) con el valor <00X>. La señal RESET de la PROM debe proceder de la señal INIT, y la entrada CE de la señal DONE.

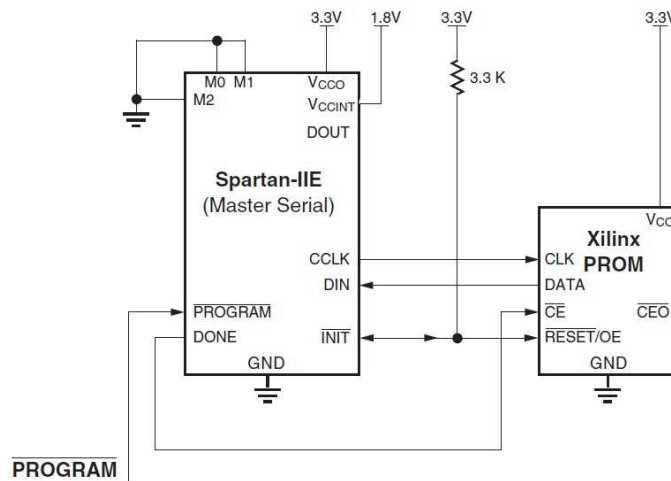


FIGURA 22: CONFIGURACIÓN MASTER SERIAL

El fichero que configura el comportamiento de la FPGA se trata de un fichero “*.bit”, que se obtendrá mediante un paquete software. Partiendo de este fichero se generará otro fichero “*.mcs” que será el que lleve cargado la FPGA. Ambos ficheros deben ser descargados al dispositivo Xilinx desde el ordenador a través del programador JTAG. La colocación de estas señales sobre el circuito será la siguiente, tal y como indica el fabricante:

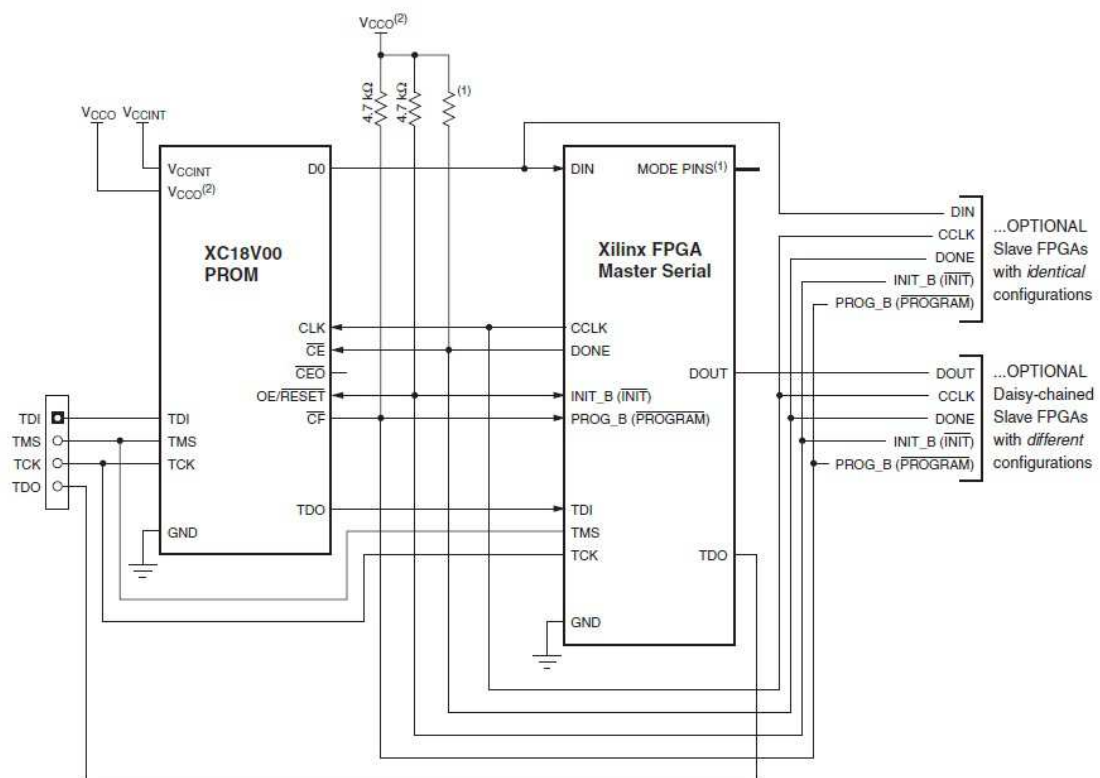


FIGURA 23 : SEÑALES DEL JTAG EN MODO MASTER SERIAL

Por último, dependiendo del modo de configuración seleccionado, el CCLK puede ser una salida generada por la FPGA, o puede ser generada externamente, y proporcionado a la FPGA como una entrada.

En el modo Master Serial, el CCLK es una salida y es llevada desde la FPGA a la PROM de Xilinx:

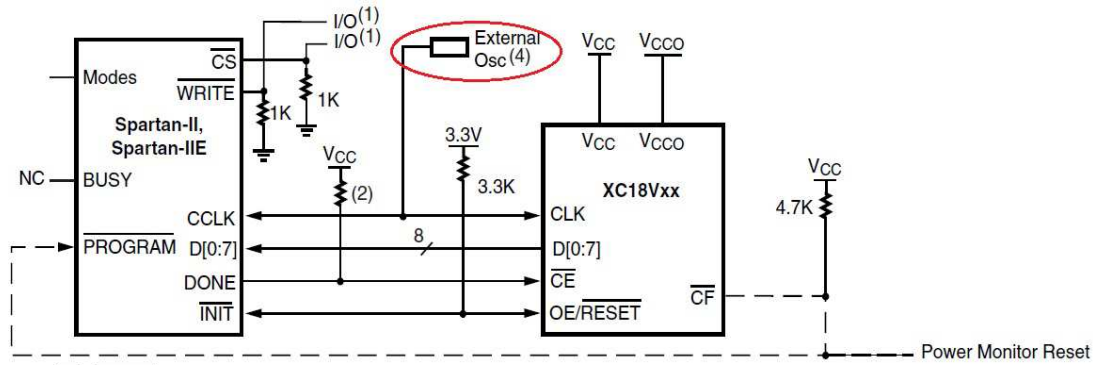


FIGURA 24: RELOJ EXTERNO

En cuanto a la distribución de reloj ya mencionada, se debe destacar que en la familia Spartan-IIE existen recursos de enrutamiento global (*Global Routing*) que distribuyen relojes y otras señales por todo el dispositivo.

Se dispone de cuatro pines de reloj dedicados, uno adyacente a cada uno de los búferes globales. La entrada del búfer global se selecciona ya sea desde estos pads o desde las señales en la ruta de propósito general. En la siguiente figura se muestra la red global de distribución de reloj en la FPGA Spartan-IIE:

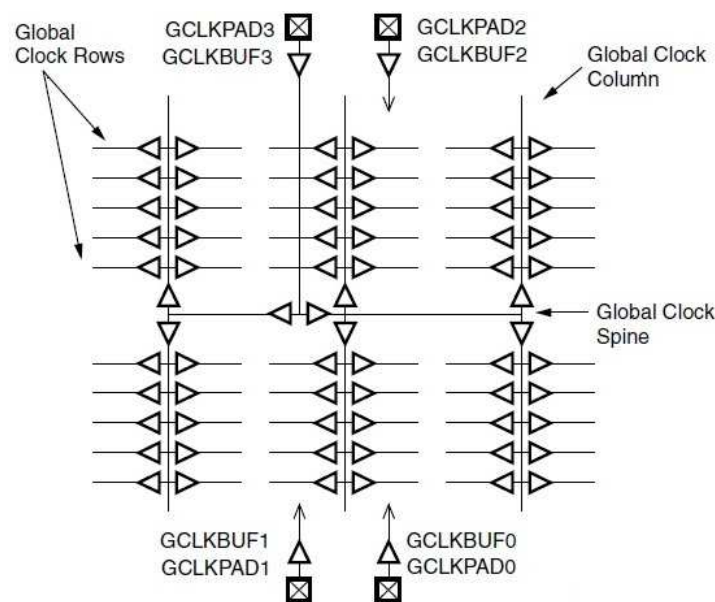


FIGURA 25 : RED DE DISTRIBUCIÓN DEL RELOJ GLOBAL

Para la configuración de la FPGA en este Proyecto Fin de Carrera, se va a emplear como reloj externo, el Crystal Oscillator VCC1 de Vectron que es un oscilador de cuarzo estabilizado que genera una onda cuadrada con salida CMOS, y que opera con tensiones de 1.8, 2.5, 3.3 o 5.0V.

El oscilador VCC1 utiliza el armónico fundamental o el tercer armónico, lo que da como resultado un bajo jitter. (typically 0.5ps rms in the 12kHz to 20mHz band). Además se presenta como un circuito integrado herméticamente cerrado, lo que supone una mejor fiabilidad y reducción de coste. Éste será el oscilador externo que se empleará en este diseño:

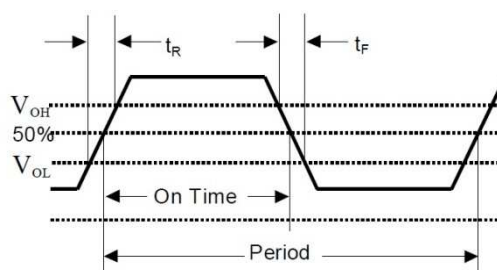


FIGURA 26 : FORMA DE ONDA DE LA SALIDA DEL OSCILADOR

Dichas señales de reloj deben llevarse a los pines dedicados de la FPGA que especifica el fabricante en sus hojas de características:

Clock	Bank	P		N	
		Pin	Name	Pin	Name
GCK0	4	AA12	GCK0, I	Y12	I/O (DLL), L#P
GCK1	5	AB12	GCK1, I	AB11	I/O (DLL), L#N
GCK2	1	A11	GCK2, I	A12	I/O (DLL), L#P
GCK3	0	C11	GCK3, I	B11	I/O (DLL), L#N

TABLA 17: PINES DEDICADOS PARA SEÑALES DE RELOJ

En cuanto a las señales de alimentación ($V_{ccint}=1.8V$ $V_{cco}=3.3V$) también especifica lo siguiente extraído de la información del fabricante:

VCCINT Pins								
-	-	E5	E18	F6	F17	G7	G8	G15
G16	H7	H16	R7	R16	T17	T8	T8	T16
U6	U17	V5	V18	-	-	-	-	-
VCCO Bank 0 Pins								
F7	F8	G9	G10	-	-	-	-	-
VCCO Bank 1 Pins								
F15	F16	G13	G14	-	-	-	-	-
VCCO Bank 2 Pins								
G17	H17	J16	K16	-	-	-	-	-
VCCO Bank 3 Pins								
N16	P16	R17	T17	-	-	-	-	-
VCCO Bank 4 Pins								
T13	T14	U15	U16	-	-	-	-	-
VCCO Bank 5 Pins								
T9	T10	U7	U8					
VCCO Bank 6 Pins								
N7	P7	R6	T6					
VCCO Bank 7 Pins								
G6	H6	J7	K7					
GND Pins								
A1	A2	A22	B1	B2	B21	C3	C20	G11
G12	J9	J10	J11	J12	J13	J14	K9	K10
K11	K12	K13	K14	L7	L9	L10	L11	L12
L13	L14	L16	M7	M9	M10	M11	M12	M13
M14	M16	N9	N10	N11	N12	N13	N14	P9
P10	P11	P12	P13	P14	T11	T12	Y20	Y3
	AA2		AA21		AB1	AB22	-	-
Not Connected Pins								
A2	B1	D4	D19	W4	W19	Y4	AA4	AA22

TABLA 18: PINES DEDICADOS PARA SEÑALES DE ALIMENTACION

Los dispositivos están disponibles en una variedad de encapsulados.

Package	Leads	Type	Maximun I/O	Lead Pitch (mm)	Footprint Area(mm)	Height (mm)	Mass (g)
TQ144 / TQG144	144	Thin Quad Flat Pack (TQFP)	102	0.5	22 x 22	1.60	1.4
PQ208/PQG208	208	Plastic Quad Flat Pack (PQFP)	146	0.5	30.6 x 30.6	3.70	5.3
FT256 / FTG256	256	Fine-pitch Thin Ball Grid Array (FBGA)	182	1.0	17 x 17	1.55	1.0
FG456 / FGG456	456	Fine-pitch Ball Grid Array (FBGA)	329	1.0	23 X 23	2.60	2.2
FG676 / FGG676	676	Fine-pitch Ball Grid Array (FBGA)	514	1.0	27 X 27	2.60	3.1

TABLA 19: ENCAPSULADOS DISPONIBLES DE LA FPGA

En definitiva se elige el integrado FG456, debido a comunalidades de otros módulos y stock de la empresa.

3.2.2.2. PUERTO SERIE

Según se comento en el punto 3.1.1.2, dado que los niveles de tensión en el estándar RS-232 no son compatibles con los niveles TTL, se requiere un transceiver que convierta estas tensiones de unos niveles a otros y viceversa.

Se empleará un circuito integrado de la familia de MAXIM, que convierte las señales de un puerto serie RS-232 a señales compatibles con los niveles TTL de los circuitos lógicos. El MAX3314 servirá como interfaz de transmisión y recepción para las señales RX y TX.

El circuito integrado tiene salidas para manejar niveles de voltaje del RS-232 (aproximadamente $\pm 7.5V$) que las produce a partir de un voltaje de alimentación de +5V utilizando multiplicadores de voltaje internamente en el chip con la colaboración de los condensadores externos.

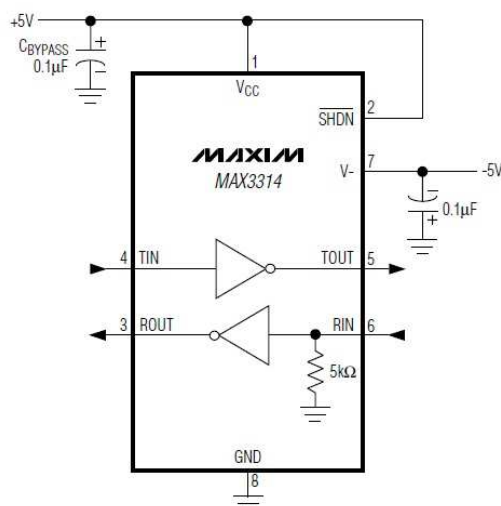


FIGURA 27 : CIRCUITO TÍPICO DE FUNCIONAMIENTO

El chip existe en diferentes encapsulados,

PART	TEMP. RANGE	PIN PACKAGE
MAX3314CKA	0°C to +70°C	8 SOT23
MAX3314CUA	0°C to +70°C	8 µMAX
MAX3314CSA	0°C to +70°C	8 SO
MAX3314EKA	-40°C to 85°C	8 SOT23
MAX3314EUA	-40°C to 85°C	8 µMAX
MAX3314ESA	-40°C to 85°C	8 SO

TABLA 20: CARACTERÍSTICAS ENCAPSULADOS MAX3314

Para este proyecto fin de carrera se elegirá el modelo MAX3314EUA debido a que ocupa menor espacio y no se debe perder de vista que el espacio en la PCB es bastante escaso. Además cumple con las especificaciones requeridas para los márgenes de temperatura de la TIC del EAE.

Una vez adaptadas las señales del puerto serie para trabajar en TTL, habrá que implementar el comportamiento de estas líneas del MAX3314 desde el código de la FPGA, para que ésta pueda interpretar el puerto serie procedente de la Tarjeta Procesadora. Para este caso se dispone ya de un bloque específico para este uso desarrollado en VHDL y que está ubicado dentro de las librerías de VHDL del propio departamento de Áreas Tecnológicas, contexto de trabajo de este Proyecto Fin de Carrera.

3.2.2.3. CONTROL DE UMBRALES

De acuerdo a lo mencionado en el punto 3.1.1.3, se va a necesitar una resistencia programable con la que configurar diversos valores de umbral.

El circuito integrado AD5290 de ANALOG DEVICES es uno de los potenciómetros digitales compactos y de alto rendimiento que presenta el mercado en la actualidad. Se puede utilizar como resistencia programable o como divisor de resistencia, realiza el mismo ajuste electrónico que los potenciómetros convencionales con una resolución mejorada y con mejor estabilidad de temperatura.

Se manejan a través de una interfaz serie SPI (Serial Peripheral Interface). La arquitectura del AD5290 es la siguiente:

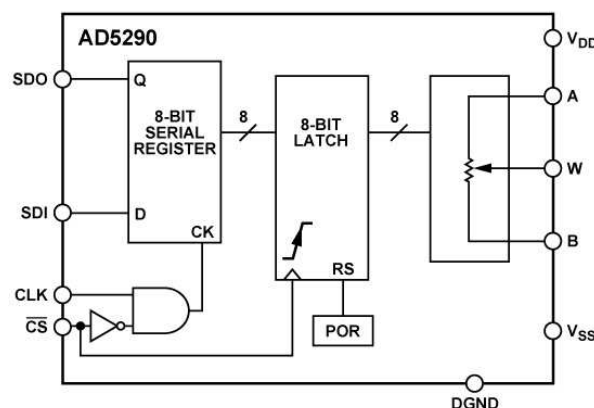


FIGURA 28 : ARQUITECTURA DEL AD5290

El bus SPI incluye una línea de reloj, línea de dato de entrada, línea de dato de salida y una entrada de “chip select” que conecta o desconecta la operación del dispositivo.

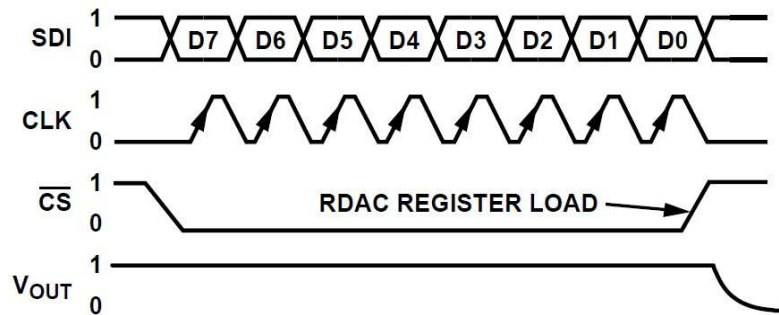


FIGURA 29 : CRONOGRAMA SEÑALES AD5290

Uno de los modos de funcionamiento de este potenciómetro digital se trata del modo “Voltage Output Operation” que consiste en implementar el siguiente circuito:

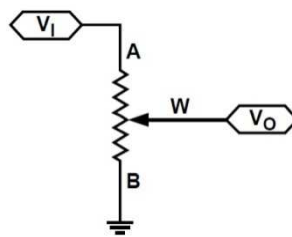


FIGURA 30 : CIRCUITO EQUIVALENTE AD5290

Cada LSB de tensión es igual a la tensión aplicada a través del terminal A y el terminal B, dividido por las 256 posiciones del divisor del potenciómetro. La ecuación general que define la tensión de salida (V_W) con respecto a la tierra para cualquier voltaje de entrada válida en el terminal A y el terminal B es:

$$V_W(D) = \frac{D}{256} \times V_A + \frac{256 - D}{256} \times V_B$$

Para el caso de la TIC objeto de este PFC, será necesario implementar las señales SDI, CLK y CS y así poder obtener en el pin W la tensión de umbral configurada según la frecuencia determinada por la Tarjeta Procesadora. Cabe destacar que la frecuencia de reloj para el funcionamiento de este chip es de 4MHz, (según especifica el fabricante) lo que supone que se deba adecuar el reloj del sistema mediante un divisor de frecuencia para generar correctamente la señal CLK. El planteamiento será implementar el comportamiento de estas líneas de entrada al AD5290 desde el código de la FPGA.

3.2.2.4. CONTROL DE ATENUACIONES

En el estudio preliminar del diseño, para el caso de los atenuadores MAADSS0013, se determinó que no era necesario implementar ningún bloque hardware para configurar su funcionamiento. Desde la lógica programable se le indicará la palabra de 4 bits a la que programarse.

Pero en cuanto a la etapa RF, tal y como se indicaba en el punto 3.1.1.4 se trata de proporcionar al atenuador HMC-C053 un valor analógico entre 0 y -3V. El fabricante especifica tablas mediante las cuales programar con este valor de tensión el valor de atenuación según frecuencias de trabajo.

La Tarjeta Procesadora será la encargada de disponer de esas tablas, para la TIC es transparente qué valor se programe según la frecuencia. Tan solo debe ser capaz de capturar el valor al que programarse y traducirlo en un valor entre 0 y -3V.

La solución óptima para ello sería implementar un circuito conversor digital analógico, que para determinado valor digital la TIC programe el valor correspondiente en tensión. Solo habrá que configurar una tabla de valores de tensión para que la TP pueda administrar el valor que considere adecuado.

Los circuitos conversores digital-analógicos son empleados en casos en que la salida de un circuito digital debe entregar una tensión o corriente para activar algún dispositivo analógico. Es decir que la salida de un circuito digital puede convertirse en una señal de control para ajustar casi cualquier variable analógica.

Los parámetros clave que definen un DAC (digital-to-analog converter) son la resolución que viene dada por el número de bits de entrada, el tiempo de conversión, tensión de referencia, tensión de alimentación, y básicamente su estructura interna.

En resumen, se puede decir que hay dos tipos de DAC. Los DAC con salida en corriente, que como su propio nombre sugiere, proporciona corriente como señal de salida y los DAC con salida en tensión que convierten la señal de corriente en una señal de tensión. Estos últimos son más lentos que los DAC con salida en corriente debido al retardo introducido por la conversión a tensión. Lo habitual es necesitar una señal analógica como el caso de este control de atenuaciones, por lo tanto una práctica común es emplear DAC con salida en corriente para beneficiarse de la velocidad de conversión y realizar la conversión a tensión mediante amplificadores operacionales externamente.

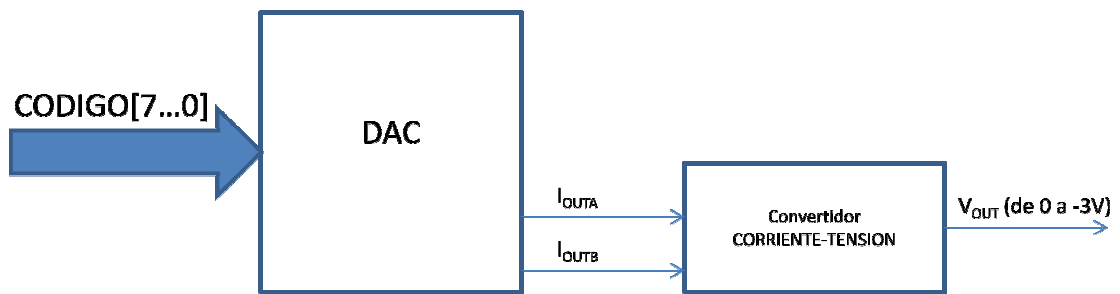


FIGURA 31 : DIAGRAMA CIRCUITO CONVERSOR DIGITAL-ANALÓGICO

Los convertidores digital-analógicos están disponibles como circuitos integrados. El THS5671A de TEXAS INSTRUMENTS es un convertidor muy utilizado en el contexto de trabajo para aplicaciones de este tipo. Y una de las ventajas que supone esto, es que se dispone del chip en stock. Por lo tanto, la opción de utilizar este integrado supone una elección buena porque abarata costes y ahorra tiempo, pero además se trata de un dispositivo de alto rendimiento y baja potencia de disipación.

El THS5671A tiene una resolución de 14 bits, y cuenta con tensión de referencia interna. Tiene un buen rendimiento con una velocidad de 125MSPS. Se alimenta a 5V y sus salidas pueden ser ajustadas en un rango desde 2 hasta 20mA manteniendo un adecuado comportamiento dinámico. La estructura interna del integrado y su encapsulado será la siguiente:

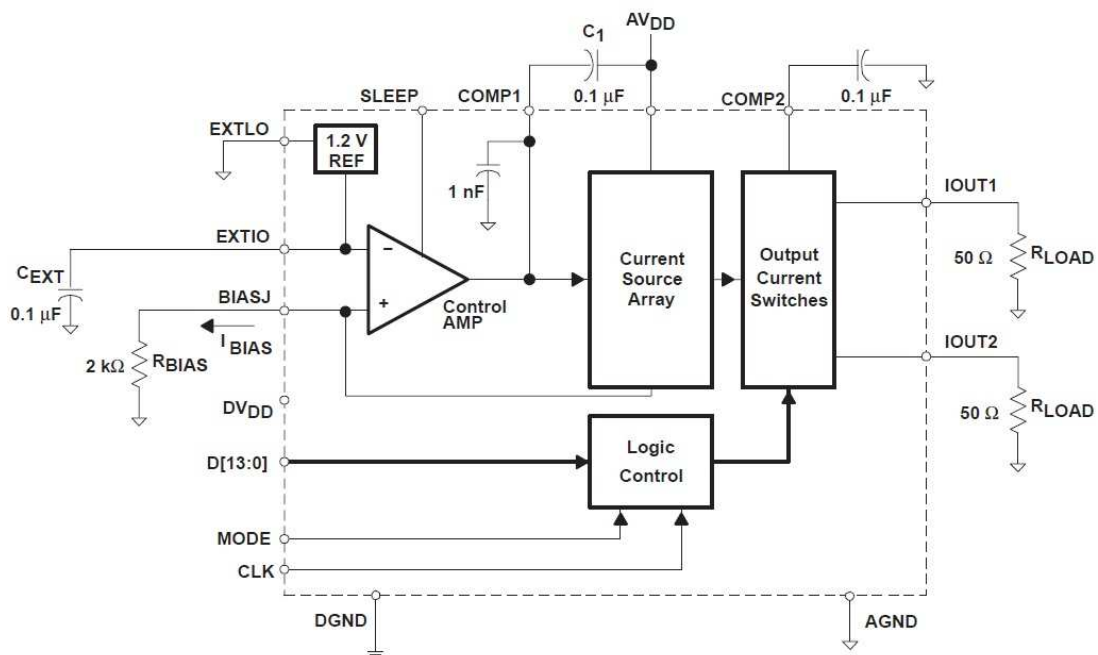


FIGURA 32 : ESTRUCTURA INTERNA DE THS5671A

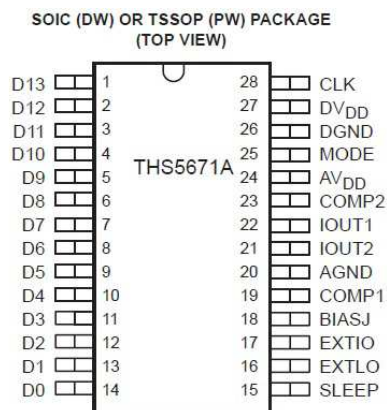


FIGURA 33 : PINOUT AD9754

Para implementar el bloque “Convertidor CORRIENTE-TENSIÓN” de la figura 31, se utilizará una estructura de un amplificador de instrumentación. Estos dispositivos están diseñados para tener una alta impedancia de entrada y alto rechazo al modo común (CMRR), y además la operación que realice es la diferencia entre sus dos entradas multiplicadas por un factor de ganancia. Lo interesante de esto, es que además de la conversión de salida en corriente a salida en tensión, además se puede jugar con este factor por si hiciera falta para futuras aplicaciones o bien dejar ganancia “1”.

La estructura es la siguiente:

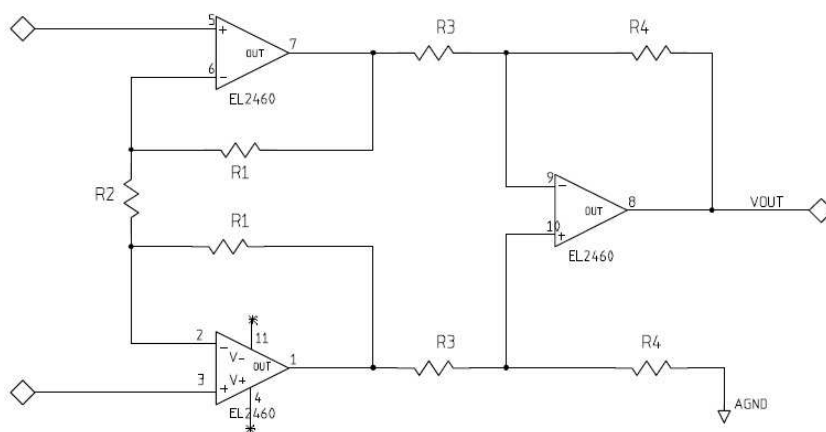


FIGURA 34 : ESQUEMA ELÉCTRICO DE UN AMPLIFICADOR DE INSTRUMENTACIÓN

Por el mismo motivo que se eligió el DAC, se empleará el EL2460 para esta estructura debido a que se trata de un componente que se encuentra en stock en el almacén interno de INDRA SISTEMAS S.A.

En definitiva, el control de las atenuaciones atenderá al siguiente esquema:

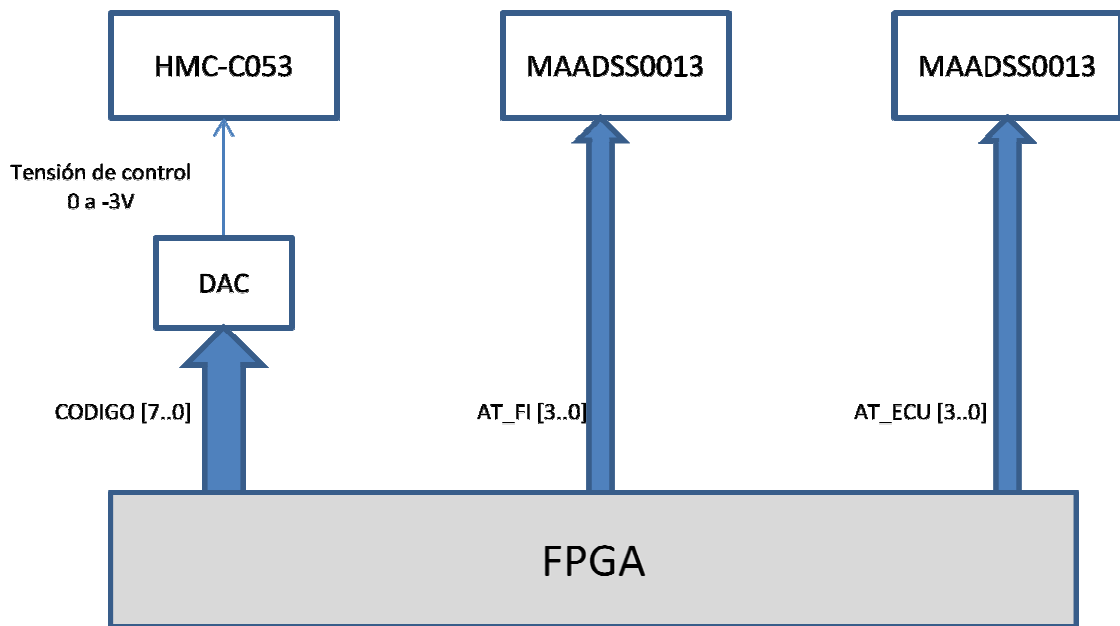


FIGURA 35 : ESQUEMA CONTROL DE ATENUACIONES

3.2.2.5. ADAPTACIÓN DE NIVELES

Como se mencionó en el punto 3.1.1.6 es necesario adaptar los niveles TTL recibidos/transmitidos desde/hacia el exterior de la TIC.

Se decide emplear el circuito integrado 74LVX3245 de ON Semiconductor que es un buffer que se utiliza para adaptar los niveles de las señales. Este dispositivo se trata de un buffer bidireccional que servirá para adaptar los niveles de las señales hacia/desde la FPGA. Además este buffer sirve para proteger a la FPGA de posibles averías debido a picos de tensión.

Recibe dos alimentaciones: V_{CCA} y V_{CCB} que son las que determinan el nivel de los puertos de datos A y B. El puerto A es un puerto dedicado a niveles de 3V, mientras que el puerto B es configurable y admite tensiones entre 3V y 5V. Este dispositivo cuenta con una señal denominada Transmit/Receive (T/\bar{R}) que determina la dirección del flujo de los datos, es decir que se puede crear una configuración donde las señales del puerto A sean entradas y las señales del puerto B sean salidas, o viceversa.

Según se especifica en el datasheet del circuito, se podrá configurar al chip para que funcione en las dos modalidades configurando la señal Transmit/Receive (T/\bar{R}). Las especificaciones que proporciona el fabricante son las siguientes:

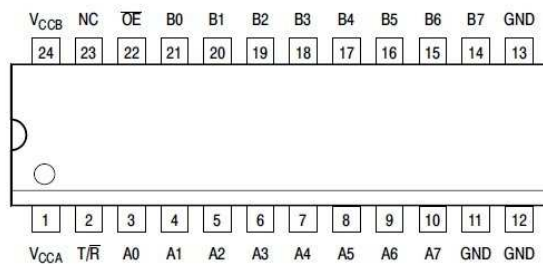


FIGURA 36 : PINOUT DEL CIRCUITO INTEGRADO 72LVX3245

Pins	Function
\overline{OE}	Output Enable Input
T/\overline{R}	Transmit/Receive Input
$A0 - A7$	Side A 3-State Inputs or 3-State Outputs
$B0 - B7$	Side B 3-State Inputs os 3-State Outputs

TABLA 21: DESCRIPCION DE PINES DE 72LVX3245

INPUTS		OPERATIN MODE
\overline{OE}	T/\overline{R}	
L	L	B Data to A Bus
L	H	A Data to B Bus
H	X	Z

TABLA 22: TABLA DE FUNCIONAMIENTO DEL 72LVX3245

El circuito está disponible en varios encapsulado, teniendo en cuenta la cantidad de buffers que serán necesarios, es de necesidad emplear el integrado con el encapsulado más pequeño que se pueda.

Se realizará la agrupación de señales, según la adaptación que se necesite realizar. A continuación se muestra un esquema resumen con lo que se pretende hacer:

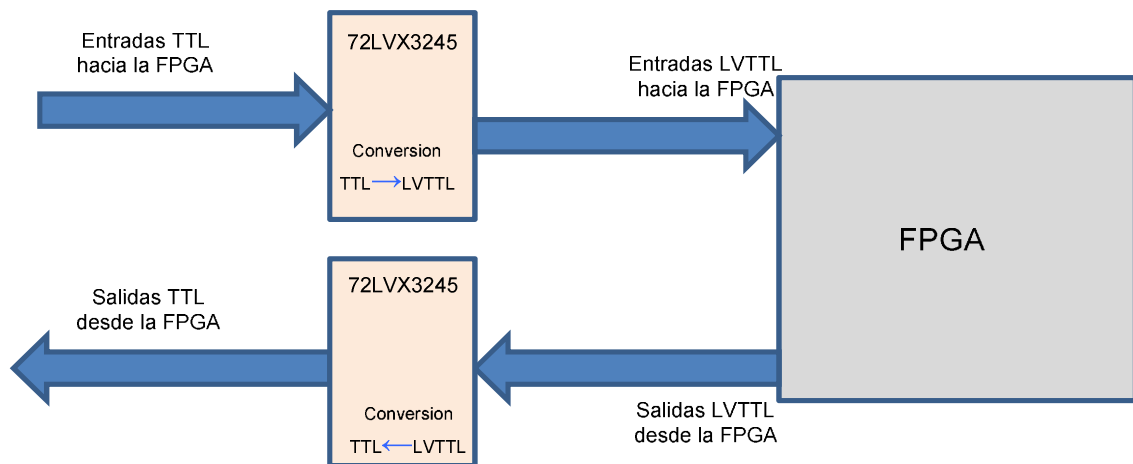


FIGURA 37 : ADAPTACIÓN DE SEÑALES DE CONTROL

Tomando como referencia la tabla 13 donde se especificaba el interfaz de señales de entrada y salida de la TIC del EAE, se puede contabilizar el número de señales que se necesitan pasar por los buffer 72LVX3245, para calcular la cantidad de integrados que se deben colocar en la PCB.

Una vez contabilizas entradas y salidas éstas resultan ser 89 entradas hacia la FPGA y 27 salidas. Lo que hace un total de 12 buffer para señales de entrada y 4 buffer para señales de salida.

3.2.2.6. FILTRADO DE ALIMENTACIONES

Como se menciona en el punto 3.1.2 resulta necesario filtrar las alimentaciones digitales que entran a la tarjeta desde el cofre del EAE por posibles interferencias entre otros módulos. Para ellos se toma la idea de utilizar los filtros EMI (Electromagnetic Interference). Los filtros EMI evitan la interferencia magnética entre dispositivos eléctricos que se encuentran cerca o quizá conectados a la misma línea de alimentación.

Se resuelve utilizar los filtros EMI de la serie BNX de Murata Manufacturing Co. Los filtros EMI de la serie BNX de Murata ofrecen una supresión de ruido en un encapsulado SMD compacto adecuado para aplicaciones industriales y comerciales. Gracias a la tecnología de encapsulados con cerámica y materiales de ferrita que presenta, los filtros EMI de la serie BNX de Murata ofrecen una amplia gama de frecuencias de 1MHz hasta 1GHz, y una tensión nominal de 50V con hasta 15A de corriente nominal en un rango de temperaturas de -40°C hasta +125°C. A continuación se muestra la descripción del funcionamiento de la serie BNX:


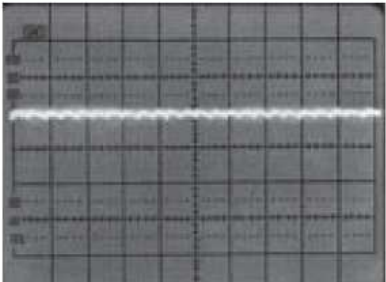
Type of Filter	EMI Suppression Effect / Description	
Without Filter	<div data-bbox="507 421 580 488">+5.0V→ 50μs/div 0.2V/div</div> 	There is high frequency noise of 0.5V maximum.
When BNX002-01 is used	<div data-bbox="507 745 580 813">+5.0V→ 50μs/div 0.2V/div</div> 	BNX002-01 can suppress most of noise.

FIGURA 38 : FUNCIONAMIENTO DE LA SERIE BNX

3.2.2.7. ADAPTACIÓN DE TENSIONES

Tal y como se comentó en apartados anteriores, es necesario realizar ciertas adaptaciones en las alimentaciones que entran a la TIC para adecuarlas al resto de elementos que van alojados en la UMA.

Además de la adaptación de las señales detalladas en el punto 3.1.3, y como se menciona en el punto 3.2.1.1, el núcleo de la FPGA se alimenta con una tensión menor que los bloques I/O. Se requiere por lo tanto a este bloque funcional añadirle la tensión de 1.8V que necesita el core. Por lo tanto, se realiza en este momento un bloque más anexo al de la figura 18:

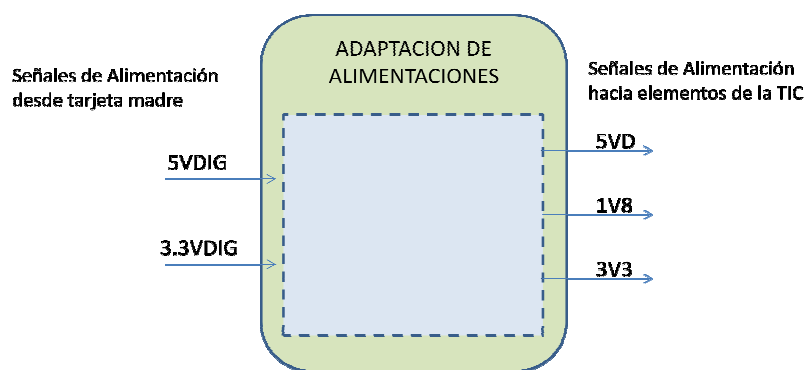


FIGURA 39 : NUEVO BLOQUE PARA ADAPTACIÓN DE TENSIONES DIGITALES

En definitiva, la relación entre tensiones será la siguiente:

TENSIÓN TARJETA MADRE	TENSIÓN REGULADA
+15V	+5V
-15V	-5V
+12V	+8V
5V5D	1V8

TABLA 23 : REGULACIÓN DE TENSIONES

Además de la regulación de tensión, se tomaba como requisito del sistema que se debía añadir una protección adicional frente a subidas de tensión y picos en los transitorios. Basándose en la figura 18, y en cuanto a las tensiones que necesitan los módulos de la UMA, se resuelve utilizar un regulador de tensión y una etapa de filtrado a la salida de la TIC. Esto será lo que se desarrollará:

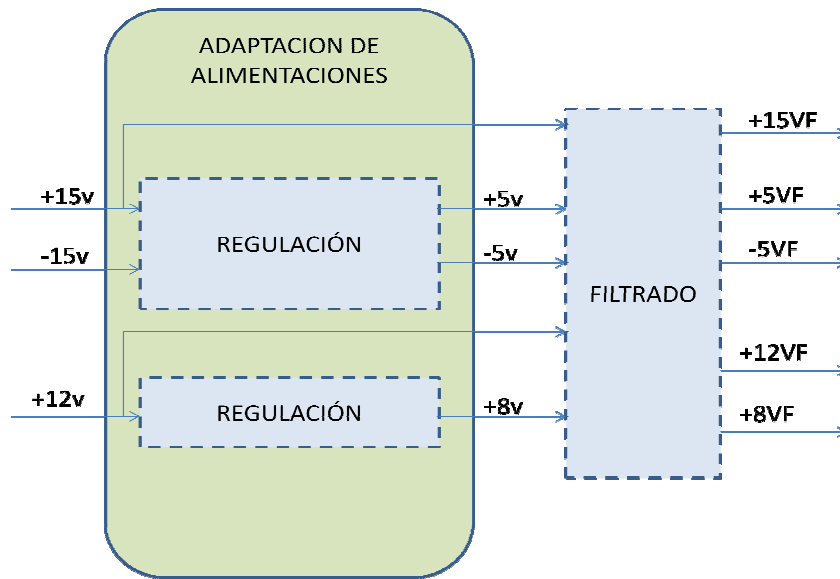


FIGURA 40 : BLOQUE ADAPTACIÓN DE TENSIONES PARA LA UMA

En cuanto a la etapa de filtrado en las tensiones analógicas de la que se hablaba en el punto, se empleará una estructura típica que se emplea en implementaciones de elementos de radiofrecuencia. Las alimentaciones una vez reguladas se pasan por un diodo supresor de transitorios de la familia SMBJ5.0A-170A de Microsemi, y después son filtradas mediante el filtro 4700-008 de Tusonix Inc, cumpliendo así con los requisitos del punto 2.2.2 (ver figura 9).

Para los requisitos de alimentación de la FPGA así como el resto de elementos digitales del sistema, se resuelve pasar las alimentaciones digitales de la tarjeta madre por el filtro BNX (detallado en el apartado anterior) y después obtener de alguna de ellas la tensión del core. El método será el siguiente:

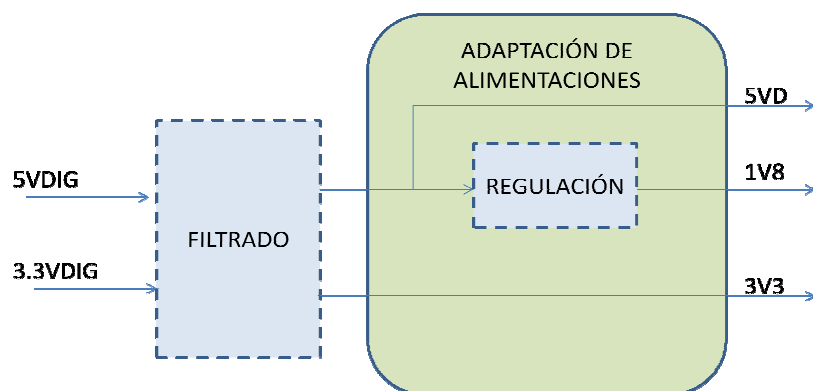


FIGURA 41 : ADAPTACIÓN DE ALIMENTACIONES DIGITALES

En ambos casos, se va a emplear la familia LM117/LM317A/LM317 para regulaciones de tensión positiva y la familia LM137/LM337 para regulaciones de tensiones negativas, ambas de *National Semiconductor*. Las características, en resumen, según indica el fabricante, de estas dos familias son las siguientes:

	LM117 / LM317A / LM317	LM137 / LM337
V_{OUT} (Para -55°C to $+150^{\circ}\text{C}$)	1,2V hasta 37V	-1,2V hasta -37V
I_{OUT}	1,5A garantizado	1,5A garantizado

TABLA 24: CARACTERÍSTICAS DE LM117/LM317

Se debe comprobar que son capaces de suministrar la corriente requerida por la unidad. El chip dispone de diferentes encapsulados.

Part Number	Suffix	Package	Output Current
LM117,LM317	K	TO-3	1.5 A
LM317A, LM317	T	TO-220	1.5 A
LM317	S	TO-263	1.5A
LM317A, LM317	EMP	SOT-223	1.0 A
LM117, LM317A, LM317	H	TO-39	0.5 A
LM117	E	LCC	0.5 A
LM317A, LM317	MDT	TO-252	0.5 A

TABLA 25 : OPCIONES DE ENCAPSULADOS PARA LM117/LM317A/LM317

Para determinar el encapsulado adecuado en función de la corriente que se precisa suministrar, será necesario realizar un análisis de consumos. Por ello se partirá de la tabla de especificaciones Tabla 5, y se añadirán los requisitos que supongan el resto de elementos de la TIC en la siguiente tabla:.

ELEMENTO		TENSIÓN	CONSUMO	UNIDADES	CONSUMO TOTAL (mA)
PART NAME	FABRICANTE	(V)	(mA)		
MAX3314EUA	MAXIM	(+5)	3,7	1	3,7
		(-5)	8,8	1	8,8
EL2460CS	ELANTEC SEMICONDUCTORS	(+5)	11	2	22
		(-5)	11	2	22
AD5290YRMZ10	ANALOG DEVICES	(+5)	20	2	40
		(-5)	20	2	40
		(+15)	0,05	2	0,1
		(-15)	0,05	2	0,1
THS5671A	TEXAS INSTRUMENTS	(+5)	30	1	30

ELEMENTO		TENSIÓN	CONSUMO	UNIDADES	CONSUMO
PART NAME	FABRICANTE	(V)	(mA)		TOTAL (mA)
		5VD	5	1	5
74LVXC3245	ON SEMICONDUCTORS	5VD	0,08	16	1,28
		3V3D	0,05	16	0,8
XC2S300E	XILINX	3V3D	400	1	400
		1V8	300	1	300
XC18V02	XILINX	3V3	25	1	25

TABLA 26: RELACIÓN DE CONSUMOS POR ALIMENTACIONES DE LA TIC

Al agrupar los consumos por alimentaciones de entrada, y teniendo en cuenta los consumos que se vieron en la tabla 7 se obtiene el siguiente el total de consumos que se indica en la siguiente tabla. Como se ha mencionado en ocasiones anteriores se procura utilizar por filosofía de empresa encapsulados de los que se dispone en stock para ahorro en costes y en tiempo. Finalmente se emplearán los siguientes integrados:

RELACION REGULADORES			
TENSIÓN TARJETA MADRE	TENSIÓN REGULADA	CONSUMO(mA)	PART NAME
+15V	+5V	1488,2	LM317T
-15V	-5V	77,5	LM337T
+12V	+8V	Despreciable	LM317EMP
5V5D	1V8	506,28	LM317T

TABLA 27: RELACIÓN REGULADORES SEGÚN CONSUMOS

3.2.2.8. OTROS

En diseños de este tipo es habitual poner los denominados condensadores de desacoplo, para aumentar el ancho de banda de supresión de ruido. Reduce el ruido de alta frecuencia en la línea de alimentación debido a la conmutación de las puertas lógicas. Debe colocarse lo más cerca posible del circuito integrado.

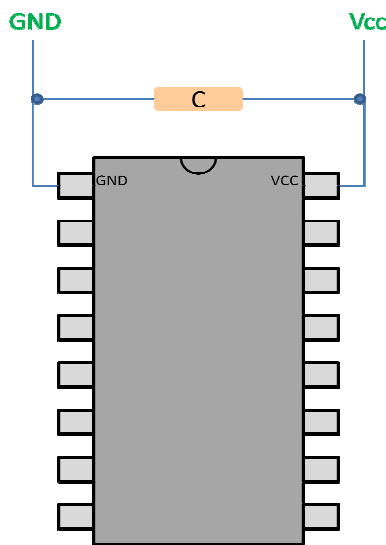


FIGURA 42 : COLOCACIÓN DE CONDENSADORES DE DESACOPLO

En cada una de las alimentaciones de los circuitos integrados que se encuentren en el diseño, se colocarán los condensadores de desacoplo adecuadamente de acuerdo a la figura 42.

3.2.3. PLANO DELINEACIÓN PRELIMINAR.

Una vez decididos los componentes, conviene plasmarlos físicamente en la PCB, para poder valorar si el espacio del que se dispone permite garantizar una mínima trazabilidad de pistas entre componentes, trabajo que se realiza en estrecha colaboración con el departamento de delineación y ruteado de tarjetas dentro de INDRA SISTEMAS S.A.

Ha de destacarse que desde el primer momento del diseño de la TIC, objeto de este Proyecto Fin de Carrera, se cuenta unas dimensiones muy limitadas, uno de los requisitos fundamentales de este diseño es lograr optimizar el crítico espacio al máximo.

Todo layout ha de cumplir ciertas reglas de diseño y una distribución de componentes adecuada. Dicha distribución de la PCB debe realizarse por zonas y agrupar éstas por funcionalidades de acuerdo a la figura 19

Los conectores estarán situados al borde la placa para facilitar las conexiones, y los buffers y etapas de salida/entrada se situaran entre los bordes de la placa y la FPGA. La FPGA estará centrada puesto que desde ella se distribuyen señales a todos los bloques funcionales.

Los conectores, tal y como se ha descrito en puntos anteriores deben adaptarse al estándar Eurocard, y por lo tanto deben ser DIN41612. Se determina de un modo preliminar que se utilicen dos conectores DIN41612 de 96 pines con los se cubren sobradamente la cantidad de entradas y salidas que se necesitan. El plano era el indicado en la figura 10 del punto 3.1, que se pasaba como requisito fundamental a la hora de diseñar la PCB la cual contaría con un espacio bastante reducido.

Lo que se pretende en todo momento es optimizar el espacio, y para ello la clave será buscar distintos componentes que posean una característica común y es que sean componentes de montaje superficial SMD (Surface-Mount Device). Esto es debido a que su tamaño es mucho más reducido y de esta forma se eliminan además tener que realizar taladros en la PCB, que impondrían también problemas de espacio.

Por lo tanto, el encapsulado de las resistencias y condensadores será de 0603 o 1206 en función de la disipación de potencia, así como se procurará elegir el resto de componentes basándose en las limitaciones de espacio.

Las pistas que llevan señales digitales se debe intentar que sean lo más cortas posibles.

Se advierte además que en cuanto a las líneas de alimentación, es preferible implementar en la PCB planos en lugar de utilizar pistas que se pueden quemar y dejar inutilizada la tarjeta.

Finalmente, en resumen, las principales reglas de diseño consisten en mantener un ancho mínimo de pistas (0.2mm para pistas de señal y 0.5mm para pistas de potencia), mantener distancia mínima entre pistas (0.2mm), y distancia mínima entre componentes (0.2mm), entre otros parámetros importantes.

3.2.3.1. CONECTORES DE LA TIC

Una vez enumeradas las líneas de señales en las tablas 13 y 14, y atendiendo a los requisitos sobre los accesos en la TIC especificados en el punto 2.1.3 (Accesos de la TIC), se decide qué conectores son los que se emplearán para la conexión física entre todos los elementos que interconecta la TIC.

- 2 conectores DIN 41612 96 Pines: interfaz entre señales de la TIC y el resto del cofre del EAE, se emplearán dos unidades.

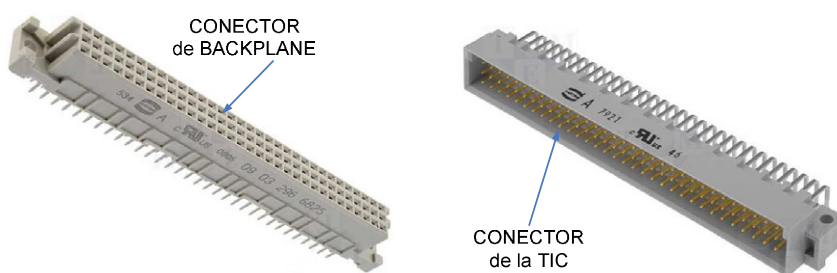


FIGURA 43 : CONECTOR DIN 41612 96 PINES

Se empleará el modelo 09 03 196 2921 del fabricante “Harting”. Este conector será el medio interfaz con la tarjeta madre y a través del cual la TIC distribuye las señales a la TP, TD y demás unidades del EAE.

- 2 conectores Micro-D 15 pines: interfaz entre señales de la TIC y determinados módulos alojados dentro de la UMA.

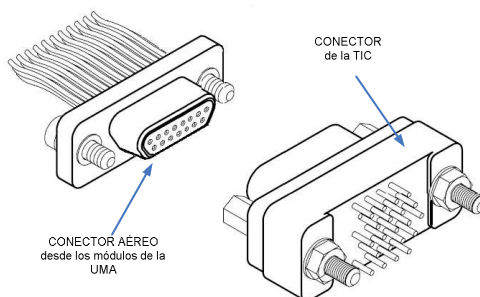


FIGURA 44 : CONECTOR MICRO-D

Se concentran en él las señales de atenuación, de autocomprobación y de alimentación de algunos de los elementos del interior de la UMA. Se empleará el modelo MK-242-015-225 de AirBorn, que presenta la huella más pequeña de los conectores micro-D encontrados en stock, característica importante debido al espacio limitado con el que se cuenta.

- 2 Conector MMCX JACK: interfaz entre señales de la TIC y determinados módulos alojados dentro de la UMA.



FIGURA 45 : CONECTOR MMCX JACK MACHO RECTO PARA PCB

Se empleará el modelo MMCX de HUBER-SUHNER. Este conector servirá para recibir desde dos detectores de la UMA las indicaciones de sobrecarga. Como se advirtió en el punto 2.1.3 estas señales se reciben a través de cables coaxiales. Estos son empleados para transmisión de señales de frecuencia haciendo que sean menos susceptibles a las interferencias. Posee dos conductores concéntricos, uno interno llamado vivo que es encargado de llevar la información, y otro exterior que sirve como referencia de tierra y retorno de corrientes.

- Tiras de pines: se puede emplear pines independientes, o en tiras.
 - Interfaz entre señales de la TIC y determinados módulos de la UMA
 - Conector JTAG: interfaz entre el PC y la FPGA.



FIGURA 46 : TIRA DE PINES PARA PCB

3.2.4. LISTA PRELIMINAR DE MATERIALES

Es este punto se hace un balance de los materiales hasta ahora mencionados y los integrados básicos que harán las funciones explicadas en puntos anteriores.

De todos ellos se ha procurado elegir materiales de los que se dispone en stock, que estén indicados para aplicaciones con espacio reducido y además que estén presentes en la Lista de Componentes Preferenciales, tal y como se indicó anteriormente.

Esta primera lista de materiales tiene con finalidad realizar una primera valoración de costes y de tiempos de suministro para anticiparse a posibles retrasos en el montaje de la PCB. Pero además tiene una utilidad importante, en el momento de implementar los esquemas eléctricos de la PCB se debe contar con la captura del componente en las bases de datos del software específico que realizará dicha tarea.

Descripción	Part Name	Fabricante	Unidades
Xilinx FPGA 2S300	XC2S300E-6FG456I	Xilinx	1
Xilinx Memoria PROM 18V02	XCV18V02VQ44I	Xilinx	1
CMOS Crystal Oscillator 40MHz	VCC-B2D-40MH0	VECTRON	1
Transceiver Compatible RS-232	MAX3314EKA	MAXIM	1
Potenciómetro Digital	AD5290YRMZ10	ANALOG DEVICES	2
Convertidor Analógico-Digital	THS5671A	TEXAS INSTRUMENTS	1
Amplificador Operacional	EL2460CS	ELANTEC SEMICONDUCTOR	2
Transceiver Dual	74LVXC3245MTC	FAIRCHILD SEMICONDUCTOR	16
Filtro supresor de Ruido	BNX002-01	MURATA	2
Regulador de Tensiones Negativas TO220	LM337T	NATIONAL SEMICONDUCTOR	1
Regulador de Tensiones Positivas T0220	LM317EMP	NATIONAL SEMICONDUCTOR	1
Regulador de Tensiones Positivas SOT223	LM317T	NATIONAL SEMICONDUCTOR	2
Filtro Supresor de Ruido	4700-008	TUSONIC INC	5
Diodo Supresor de Ruido 5V	SMBJ5.0A	MICROSEMI	2
Diodo Supresor de Ruido 12V	SMBJ12.0A	MICROSEMI	1
Diodo Supresor de Ruido 15V	SMBJ15.0A	MICROSEMI	1
Diodo Supresor de Ruido 8V	SMBJ8.0A	MICROSEMI	1
Conector MMCX	82MMCX-50-0-1	HUBER-SUHNER	2
Tiras de pines	AMP-5-826629-0	TE CONNECTIVITY	1
Conector Micro D 15 pines	MK-242-015-225	AirBorn	2
Conector DIN41612 96 pines	09 03 196 2921	Harting	2

TABLA 28: LISTA PRELIMINAR DE MATERIALES

4. DISEÑO HARDWARE.

4.1. DESARROLLO DE DISEÑO

Después de haber realizado en puntos anteriores el análisis de los requisitos y las especificaciones técnicas de la TIC objeto de este Proyecto Fin de Carrera, y tras haber estudiado las soluciones más adecuadas junto con un borrador de los materiales necesarios, se procederá a implementar el diseño.

En este apartado se describe precisamente todo este proceso de implementación, desde simbolizar todas esas especificaciones mediante un esquema electrónico hasta materializarse finalmente en una placa de circuito impreso (PCB).

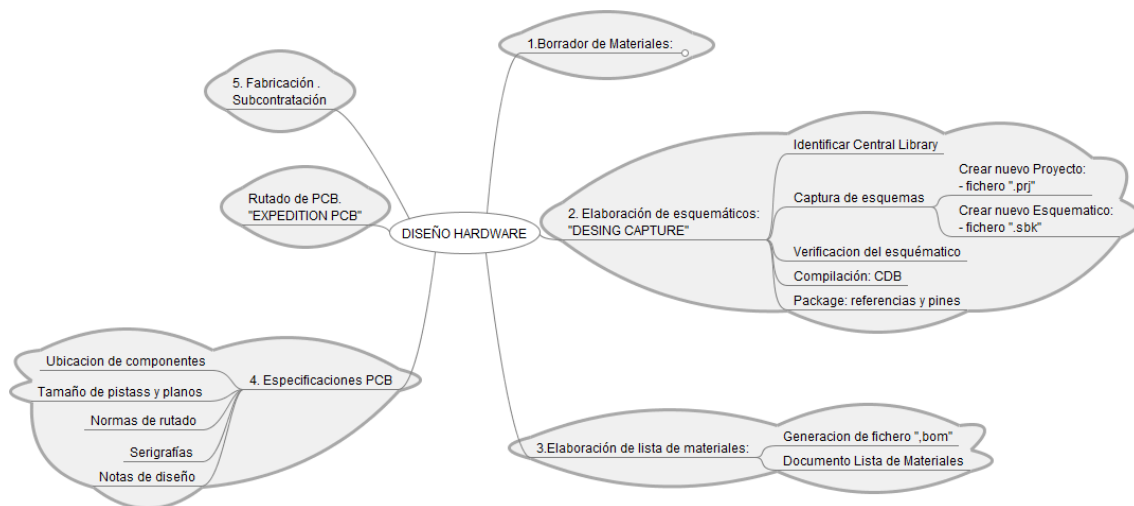


FIGURA 47 : MIND MAP TAREAS DE DISEÑO HARDWARE

Al final de este punto, se habrán realizado los esquemas eléctricos así como el resto de planos que representarán gráficamente la PCB soporte de la TIC del EAE, objeto de este proyecto Fin de Carrera.

4.2. ESQUEMAS ELÉCTRICOS.

Utilizando los medios disponibles en el contexto de trabajo, se cuenta con licencia para trabajar con el paquete de software de “*Mentor Graphics Corporation*”, porque lo que viene ya establecido tomar esta solución como herramienta software.

En concreto, para llevar a cabo el trazado de los esquemas así como el ruteado, se empleará la herramienta *Design Capture for Expedition PCB*. El entorno de Mentor Graphics supone una solución eficaz y fácil de manejar para el diseño de PCBs. Además es una práctica habitual que muchas compañías presenten archivos compartidos para reutilización de bloques de esquemáticos y además compartir librerías de símbolos, lo que hace el trabajo más eficaz y rápido.

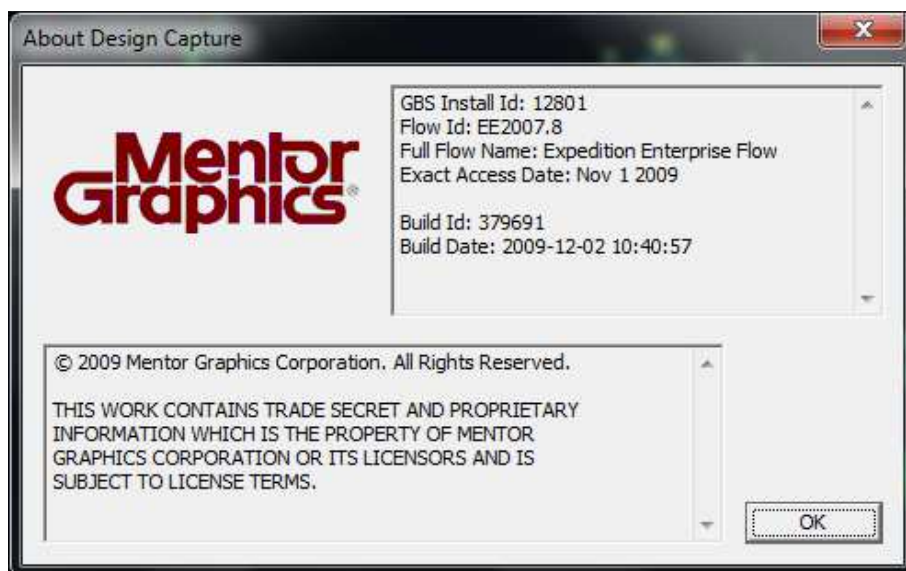


FIGURA 48 : ABOUT MENTOR DESIGN CAPTURE

Design Capture consiste en una de las herramientas que presenta este paquete software para el proceso de la creación de los esquemáticos de un diseño electrónico.

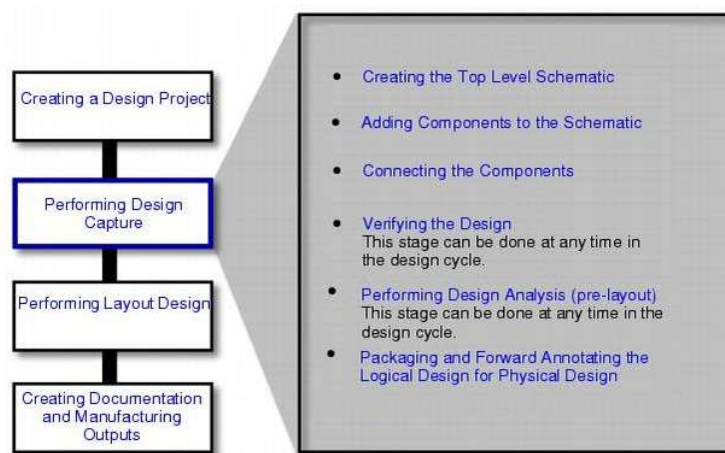


FIGURA 49 : FLUJO DE DISEÑO DESIGN CAPTURE

Expedition PCB es una de las herramientas de layout de Mentor Graphics, mediante la cual se prepara la placa de circuito impreso.

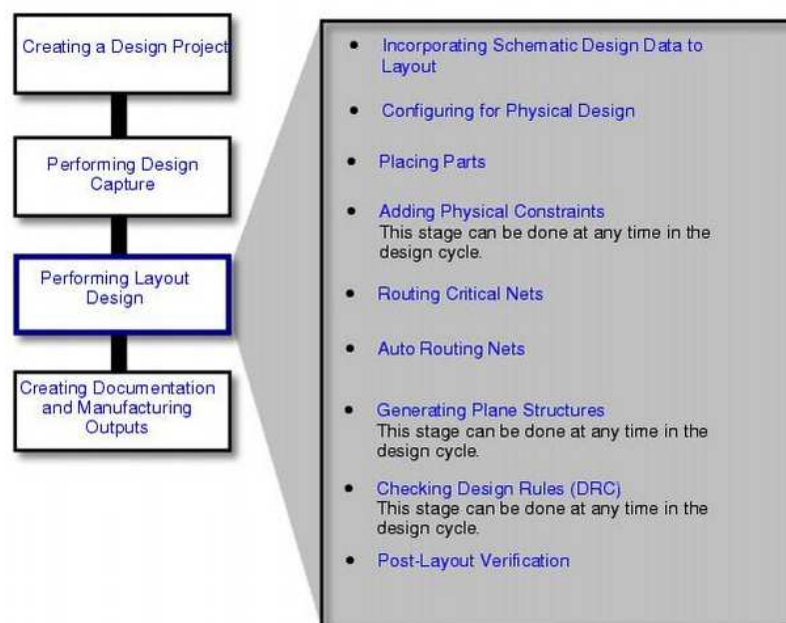


FIGURA 50 : FLUJO DE DISEÑO EXPEDITION PCB

Entre ambas se establece una conexión entre los esquemas eléctricos y el diseño de la propia PCB de manera que existe flujo de información en ambas direcciones. Por lo tanto, mediante Mentor Graphics se posibilita la captura de los esquemas eléctricos y el posterior ruteado, estando ambos enlazados.

El CDB (Common Data Base) es la versión compilada del esquemático, que contiene la información del diseño y sirve de enlace entre Design Capture y Expedition PCB.

Para trabajar con el entorno de Mentor Graphics es necesario estar familiarizado con los siguientes procesos y herramientas:

- Librería Central
- Captura del Esquemático “tic_eae_v0.sbk”
- Herramienta de Verificación
- Compilación del diseño. CDB
- Herramienta Package.
- Generación del fichero “tic_eae_v0.bom”

4.2.1. IDENTIFICACIÓN DE LIBRERÍA CENTRAL

Una Librería Central consiste en una colección de bibliotecas interrelacionadas dentro de la misma red de trabajo que proporciona una lista de componentes que se pueden trasladar al proyecto en que se esté trabajando. Tan solo es necesario indicarle la ubicación del directorio en el propio proyecto para poder empezar a utilizar los componentes de ella.

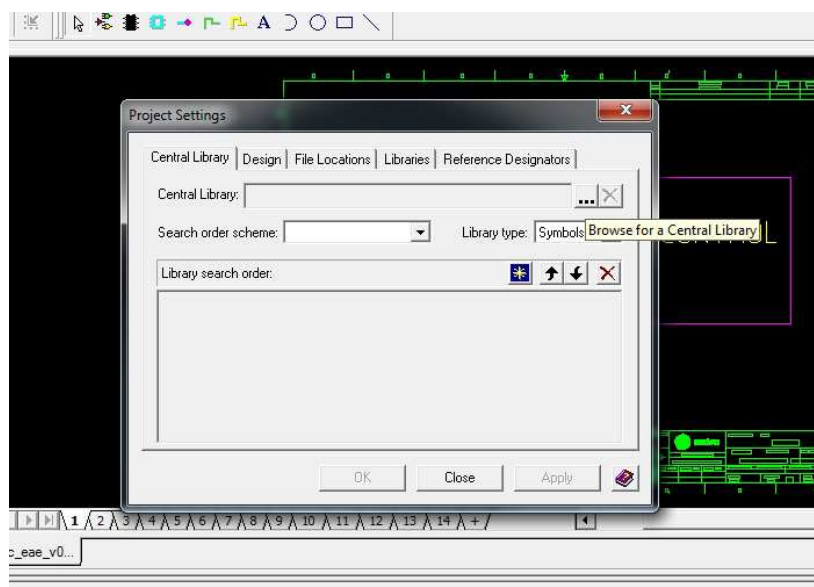


FIGURA 51 : IDENTIFICACIÓN LIBRERÍA CENTRAL

La librería almacena cada símbolo nuevo que se crea asociando su símbolo a su footprint así como el resto de las propiedades del nuevo elemento. Una vez incorporado cualquier elemento de la Librería en el proyecto en cuestión éste podrá ser editado para adecuar alguna de sus propiedades a las necesidades.

4.2.2. CAPTURA DE ESQUEMÁTICOS

El área de trabajo de Design Capture presenta la siguiente forma:

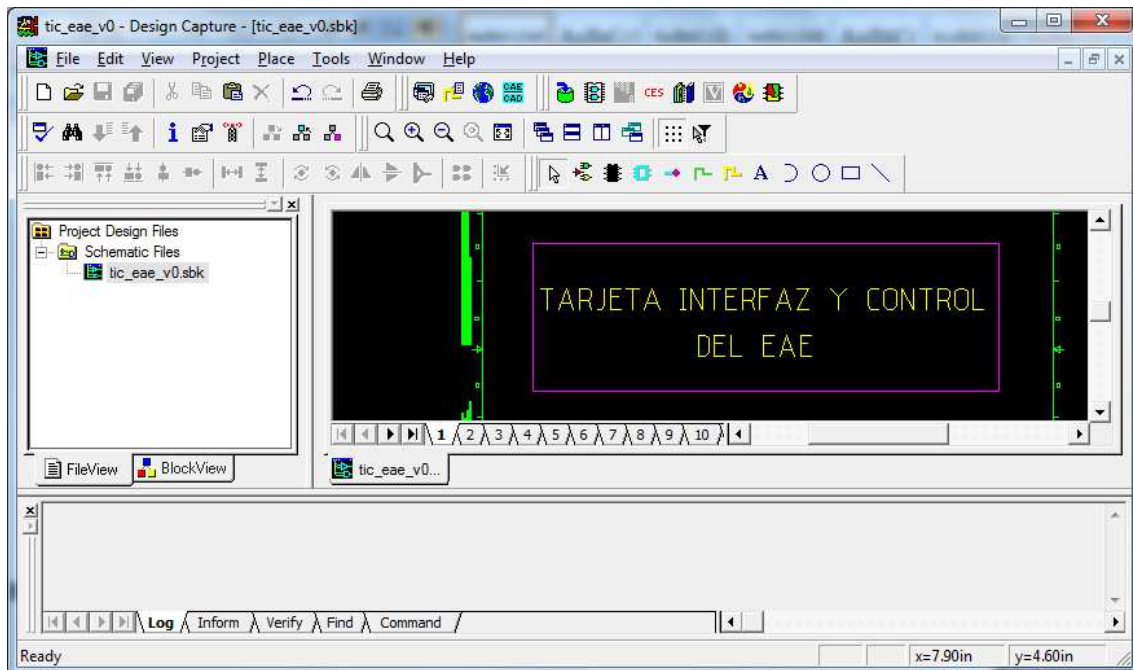


FIGURA 52 : WORKSPACE DE DESIGN CAPTURE

Lo primero que hay que hacer es crear un proyecto “tic_eae_v0.prj” y a continuación crear un fichero donde capturar el esquemático “tic_eae_v0.sbk”. Mentor Graphics recomienda ciertas consideraciones para llevar a cabo la creación de ambos, como por ejemplo que la carpeta directorio que contenga el proyecto tenga el mismo nombre o como que el proyecto esté activo mientras se modifica el fichero “tic_eae_v0.sbk”. Se debe atender a todas ellas para que se puedan utilizar satisfactoriamente el resto de herramientas del proceso de captura.

Mediante la ejecución de *Project*→*New* de la barra de herramientas comenzará un dialogo donde se especificarán la ubicación y la Librería Central que se utilizará.

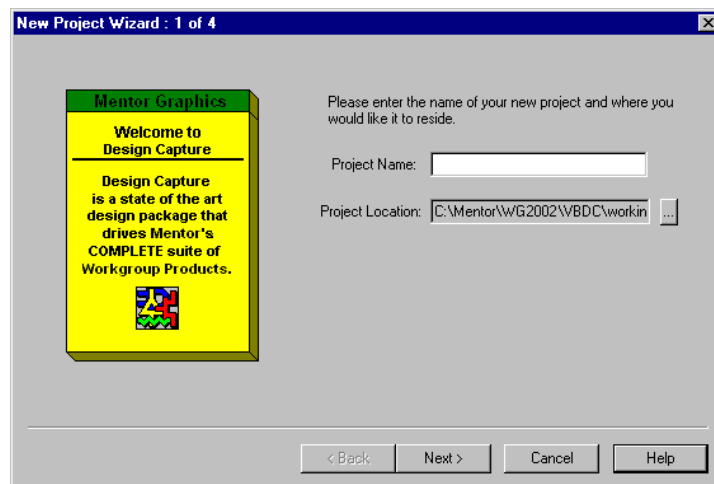


FIGURA 53 : VENTANA DE CREACIÓN DE NUEVO PROYECTO EN DESIGN CAPTURE

A continuación para crear un nuevo diseño se debe ejecutar *File*→ *New* y elegir la opción de esquemático de la lista de posibles tipos de ficheros:

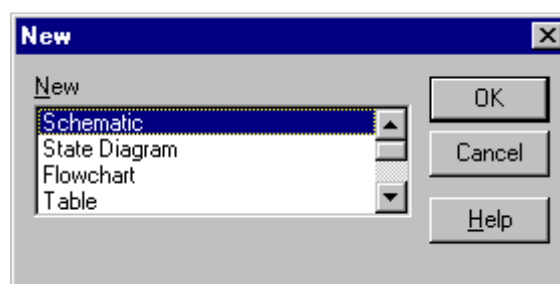


FIGURA 54 : VENTANA DIÁLOGO PARA NUEVO ESQUEMÁTICO

Una vez ya en el entorno de trabajo de Design Capture, se empezará a implementar los esquemáticos mediante la colocación de símbolos y/o bloques y realizando la conexión de los mismos (placement). Se resume a continuación ciertos rasgos importantes con los que se debe estar familiarizado para llevar a cabo la implementación del esquemático de la TIC del EAE objeto de este PFC.

Para realizar el placement se empleará el menú “Place” de la barra de herramientas:

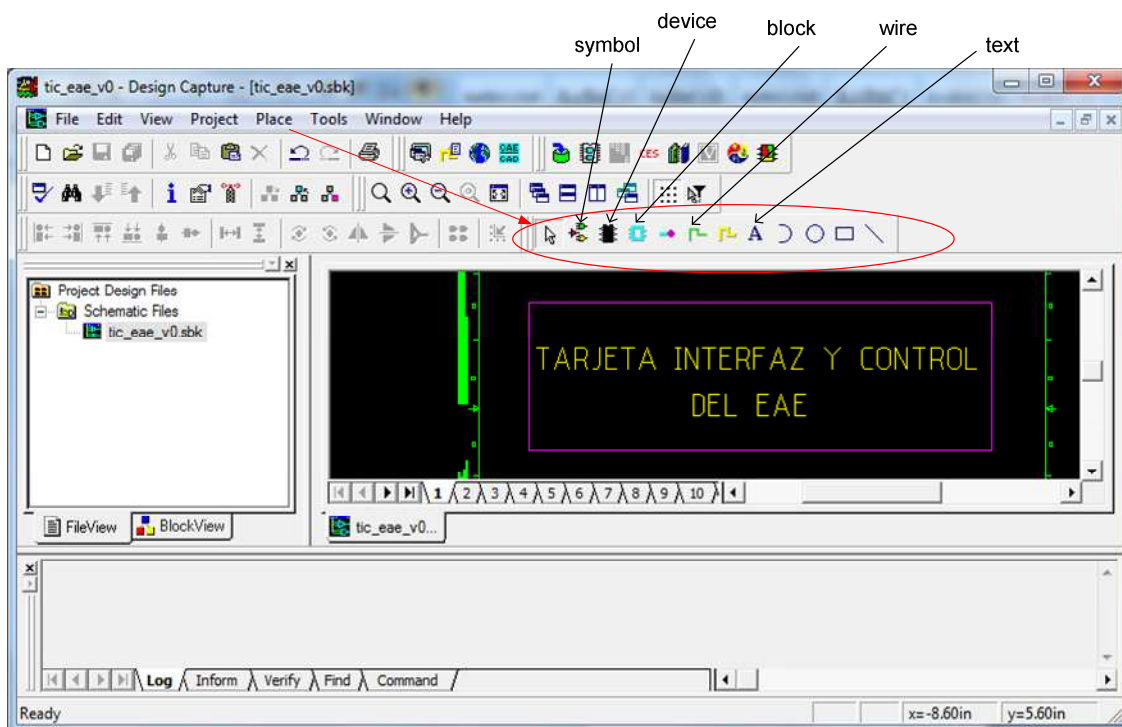


FIGURA 55 : MENÚ PLACEMENT

Un símbolo es la representación gráfica de un componente como una resistencia, un condensador, un circuito integrado, etc...presente en la tarjeta de circuito impreso. Se puede asociar a cada uno de los símbolos información adicional a la que trae ya consigo el símbolo, propiedades como por ejemplo:

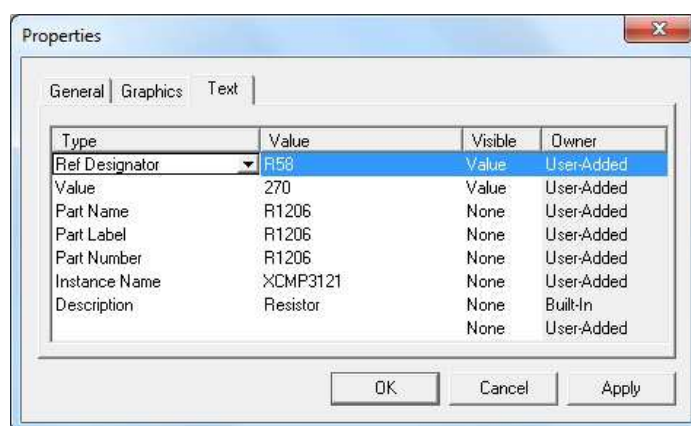


FIGURA 56 : PROPIEDADES DE LA LIBRERÍA DE UN COMPONENTE

La propiedad “Ref Designator” del ejemplo, se utiliza para identificar físicamente el elemento en la PCB (R58), y la propiedad “Value” se utiliza para asignarle un valor de resistencia (R58=270Ω).

Las uniones entre los pines cada elemento o también llamados señales o nets, se pueden seleccionar y se les puede añadir información adicional como el nombre de la net o cualquier otra propiedad que se necesite.

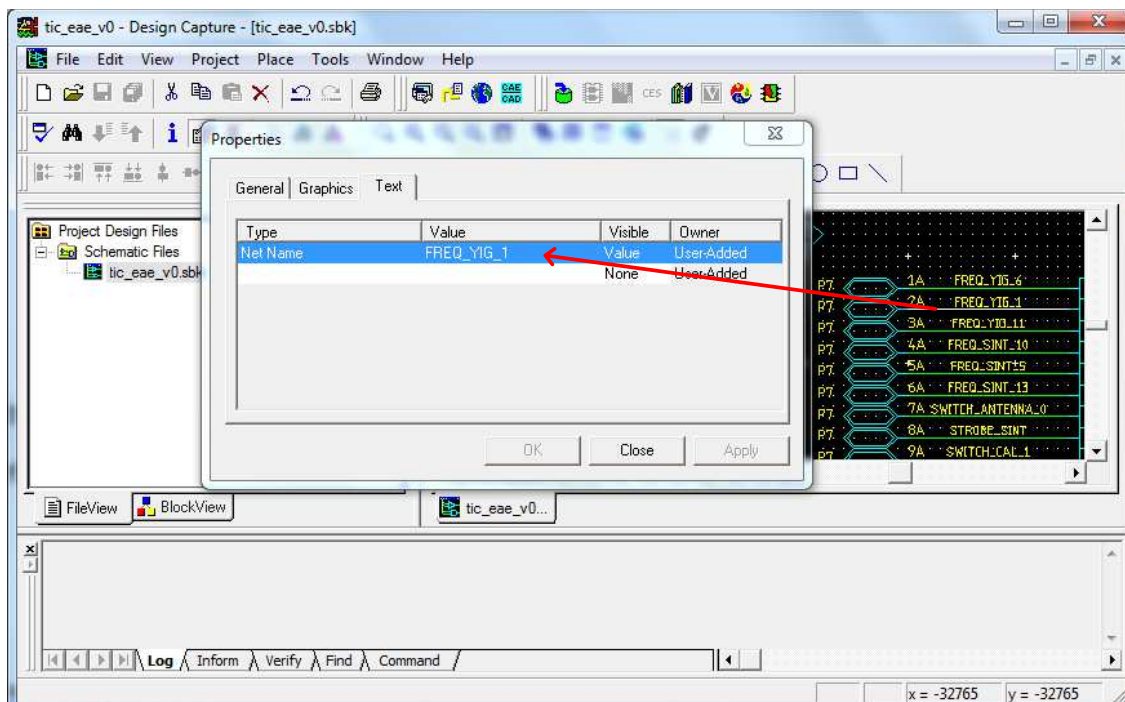


FIGURA 57 : PROPIEDADES DE UNA NET

Entre la colocación de los símbolos es importante mencionar la conexión de las nets entre las diferentes páginas que podrá presentar un esquemático.

- Conector interpagina: se utilizan para conectar wires con el mismo net a través de las páginas del esquemático.

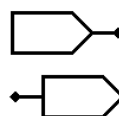


FIGURA 58 : SÍMBOLO CONECTOR INTERPÁGINA

- Conector intrapagina: En los casos donde no es práctico dibujar un wire entre dos pines que necesitan ser conectados dentro de la misma página, se utiliza estos conectores.

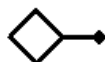


FIGURA 59 : SÍMBOLO CONECTOR INTRAPÁGINA

Ambos se encuentran clickeando en el botón “Symbol” del menú “Place”.

4.2.3. VERIFICACIÓN DEL ESQUEMÁTICO

El comando “Verify” se encuentra en el menú “Tools” y sirve para realizar algunas comprobaciones básicas sobre posibles errores en el esquemático tales como referencias repetidas, nombres de nets flotantes, pines sin conectar, etc...

Por lo tanto una vez capturado el esquemático es necesario correr el “Verify” para asegurarse que no existen errores en el proyecto.

Cuando el diseño está sin errores y listo para generar su netlist presentará la siguiente ventana inferior:

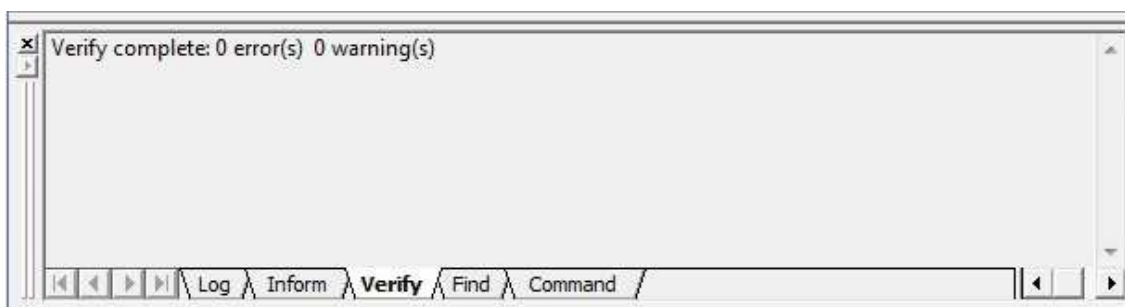


FIGURA 60 : DIÁLOGO VERIFY

4.2.4. COMPILACIÓN. CDB

Las herramientas como Expedition PCB no leen directamente de los esquemáticos de un proyecto la información del PCB. La información del esquemático debe ser escrita desde Design Capture en una base de datos llamada Common Database (CDB) antes de compartir el diseño con Expedition PCB. La CDB por lo tanto, es una base de datos en formato binario que permite intercambiar la información del diseño entre las herramientas de Mentor Graphics. Una vez que esa información estructural del esquemático ha sido escrita en la CDB, se puede transferir a Expedition PCB.

Para conseguir generar esa base de datos y una vez ejecutado satisfactoriamente el “Verify”, se debe proceder a compilar el diseño mediante la Herramienta “Compile iCDB” que se encuentra en la barra de herramientas.

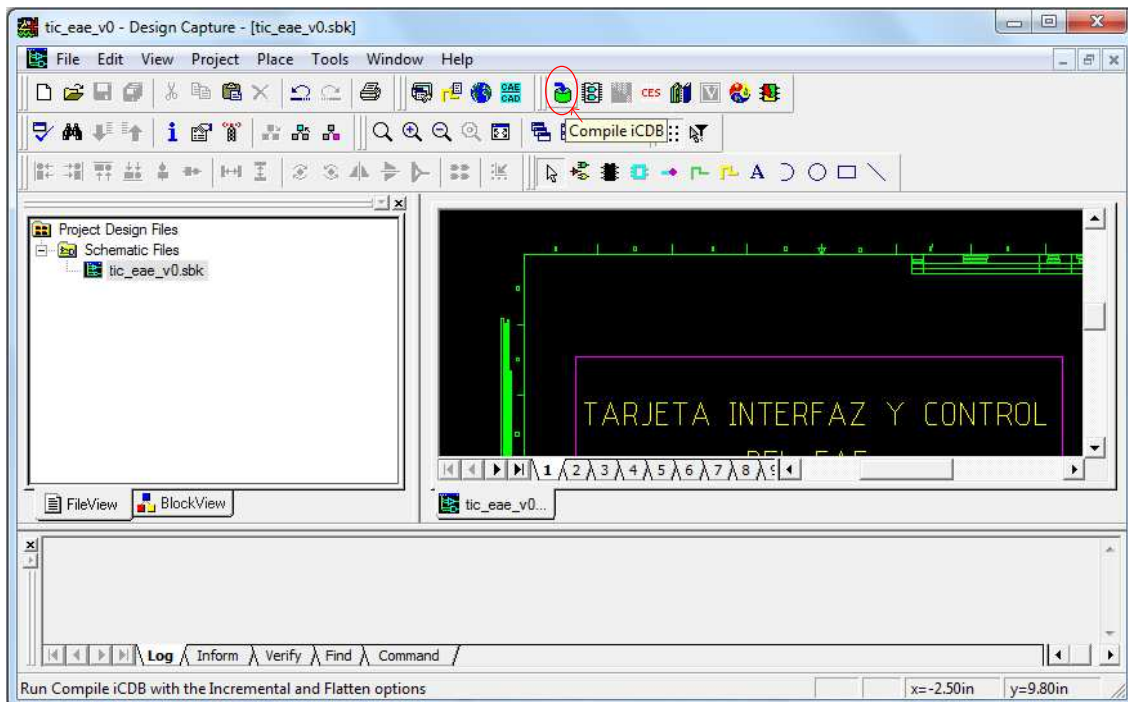


FIGURA 61: MENÚ COMPILE ICDB

El compilador crea la CDB o iCDB (integrated Common Database) mediante la lectura de los ficheros del diseño, compilándolos y escribiendo la CDB final en una localización específica del directorio del proyecto. La CDB contendrá la información estructural y de conectividad que se especificó en los esquemáticos.

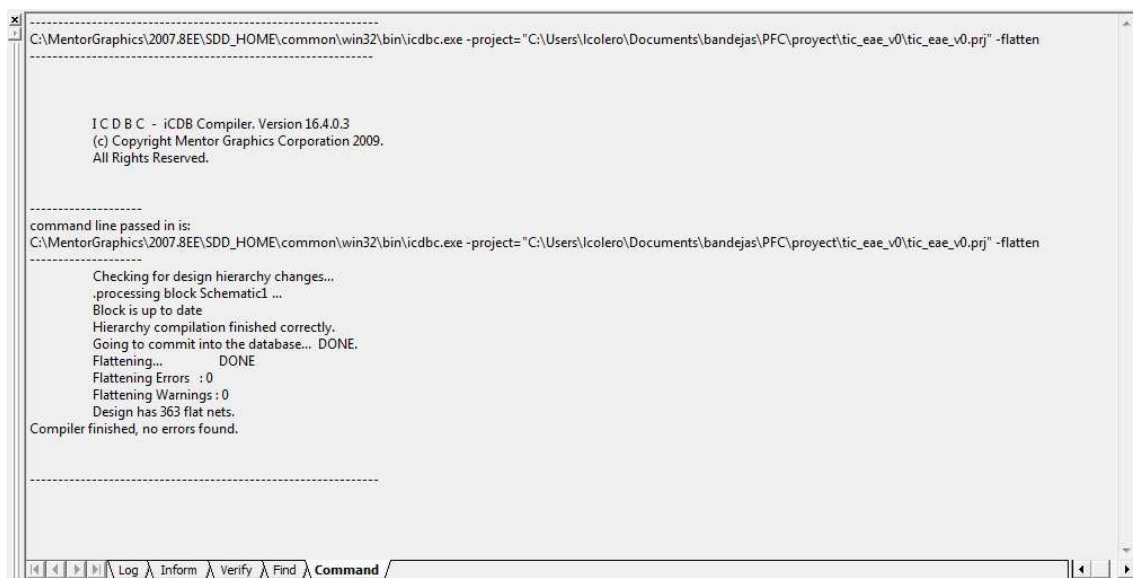


FIGURA 62 : CUADRO INFERIOR DE COMANDOS

4.2.5. PACKAGE.

En este proceso, se comprueba la CDB para asegurarse que están correctos todos los datos relevantes como referencias de símbolos o los nombres de los pines y entonces si fuera necesario asignar información de “packaging” que falte. Todo ello con el fin de preparar ya el diseño para traspasarlo a Expedition PCB.

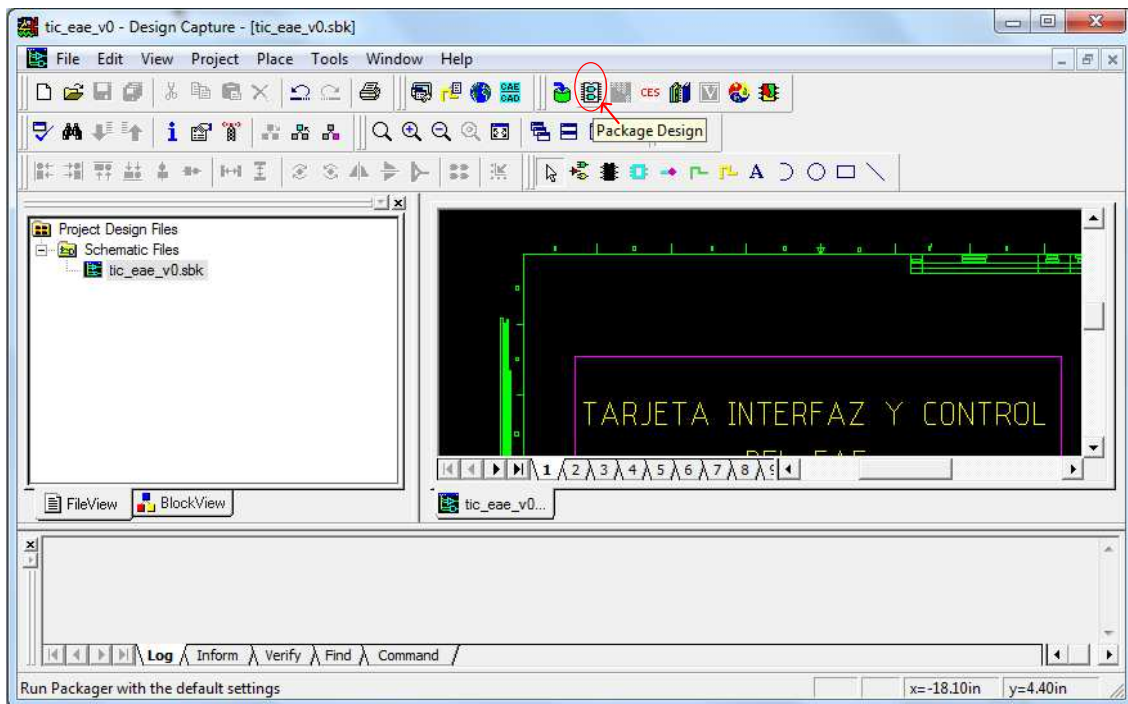


FIGURA 63 : MENÚ PACKAGE

El “Packager” lee los datos de los símbolos desde la CDB y localiza en ella la información de la parte física de dichos símbolos y asigna las propiedades de los componentes como las referencias, pin numbers, partnames, etc... de los componentes utilizados en el diseño. Al final, cada uno de los símbolos debe tener asociado toda esa información, así que si ésta no ha sido especificada en el esquemático, al ejecutar el “Packager” se asigna.

4.3. LISTA DE MATERIALES

Para cualquier compañía es muy importante tener controlado el tema de los materiales a la hora de fabricar sus sistemas. Los retrasos de los procesos de fabricación de este tipo de tarjetas como la desarrollada en este PFC, suelen darse por materiales que tarda en suministrar el fabricante. Por ello, cuanto antes se lleve a cabo la recopilación de material se estarán evitando problemas que potencialmente podrían impedir el desarrollo del resto de tareas.

Design Capture proporciona una herramienta para construir una lista de materiales a partir del esquemático del proyecto una vez compilado, utilizando la información de los “part names” de los componentes. Si un componente en la CDB contiene un part name, el componente y la información de mismo será extraída e insertada en el fichero “.bom” (Bill of Materials). Para ejecutar esta utilidad del Design Capture se deben seguir los siguientes pasos:

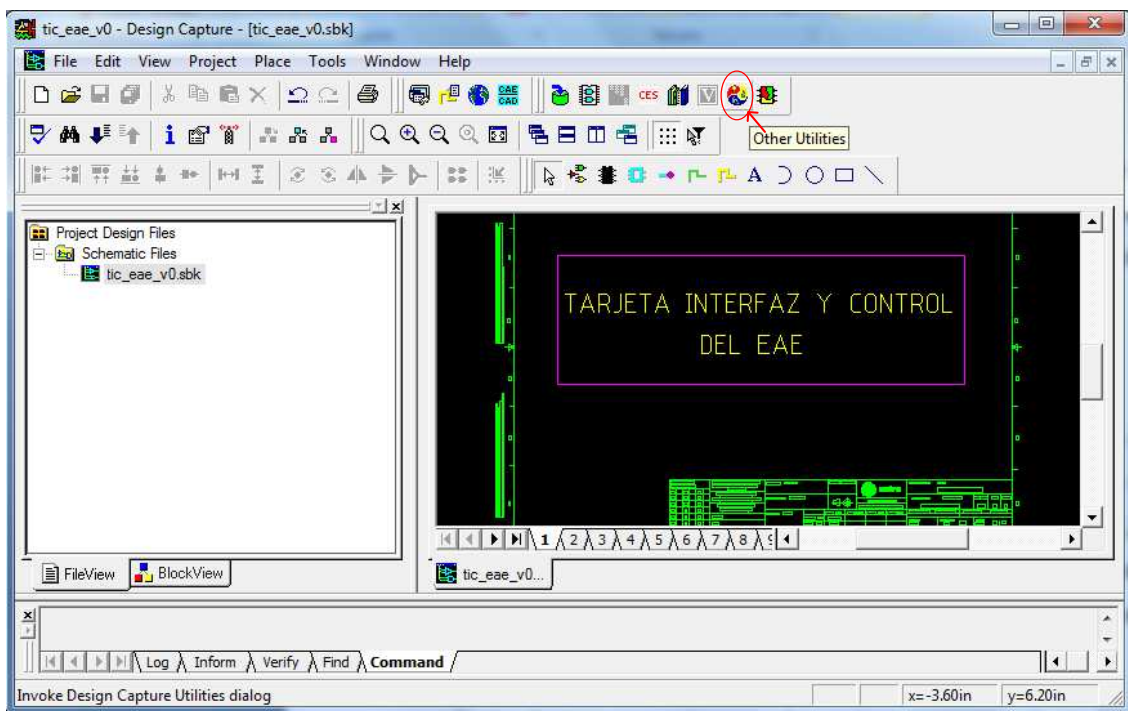


FIGURA 64 : MENÚ EXTRACCIÓN DE LISTA DE MATERIALES

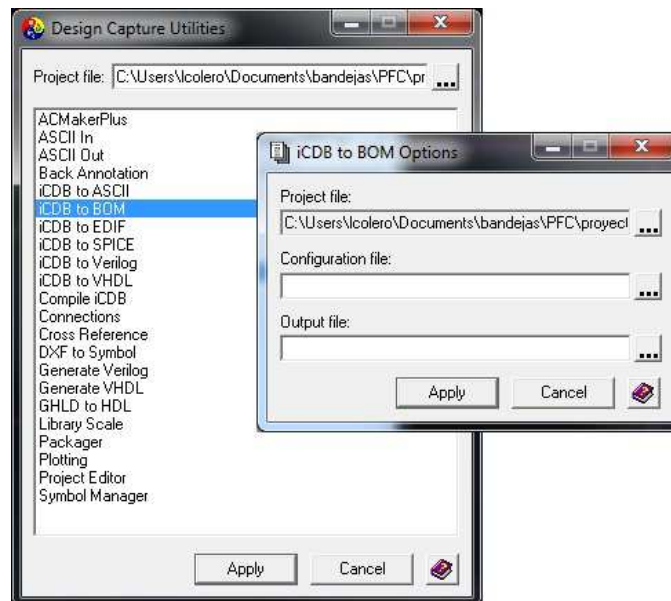


FIGURA 65 : DIÁLOGO EXTRACCION DE FICHERO *.BOM

En la ventana de dialogo que se abre seleccionar iCDB to BOM. Una vez ejecutado, se debe señalar el fichero destino de la lista de materiales construida. Después éste podrá ser abierto desde cualquier editor de texto. El contenido del fichero será como este:

Archivo: C:\Users\icolero\Documents\bandejas\PFC\project\tic_gae_v0\im 06/02/2014, 10:21:41

item	qty	part number	Value	description	references
1	2	ADS290VRMS10			U24, U42
2	1	AD9784			U12
3	2	BNX002-01			U11, U14
4	13	CTA2A	1uF	Resistor	C18, C28, C31, C43, C46, C48, C53, C94-C95, C102, C105, C109, C120
5	4	CTA2D	10uF		C5-C6, C8-C9
6	Unknown	C0603	100 pF		
7	34	C0603	100nF		
8	14	C0603	47pF		C11-C2, C29-C30, C36, C40, C44, C47, C50-C51, C54, C56, C61, C64, C74-C77, C81-C84, C91-C93, C103-C104, C106, C110-C112, C116-C118
9	6	C0603	100n		C55, C62-C63, C78
10	4	C0603			C7, C57-C60, C73
11	8	C1210			C19, C96-C97, C108
12	1	EL2460_2			C39, C41-C42, C45, C49, C52, C113-C114
13	1	LM2460_2			U26
14	1	LM317AEMP		Voltage Regulator, ADJ.	U27
15	2	LM317T		Voltage Regulator, ADJ.	U30
16	1	LM337T		Voltage Regulator, ADJ.	U1-U2
17	16	LM337T		Voltage Regulator, ADJ.	U4
18	1	MAX3314EUA			U6-U10, U16-U19, U25, U29, U31-U35
19	2	MICROD15			U3
20	2	MMCX			J10-J11
21	3	R0603	10K	Resistor	J1-U2
22	1	R0603	68	Resistor	R81, R93-R94
23	1	R0603	22	Resistor	R40
24	33	R0603	VALON	Resistor	R17
25	5	R0603	100	Resistor	R2-R9, R14, R16, R18-R22, R26-R29, R32-R36, R38-R39, R41, R51-R52, R62, R68-R70
26	8	R0603	0	Resistor	R43-R44, R47, R78-R79
27	3	R0603	50	Resistor	R1, R12, R15, R42, R61, R66-R67, R80
28	14	R0603	560	Resistor	R13, R74, R77
29	1	R0603	2K43	Resistor	R23, R25, R30-R31, R37, R45, R48-R50, R54-R56, R75-R76
30	2	R0603	1K	Resistor	R24
31	1	R0603	330	Resistor	R46, R53
32	1	R0603	4K7	Resistor	R63
33	2	R1206	270	Resistor	R82
34	3	R1206	120	Resistor	R58, R91
35	1	R1206	1K3	Resistor	R11, R57, R59
36	2	R1206	360	Resistor	R39
37	Unknown	R1206	0	Resistor	R10, R60
38	2	SMB15		Diode	D2-D3
39	1	SMB15		Diode	D5
40	1	SMB15		Diode	D4
41	1	SMB15		Diode	D1
42	Unknown	TORRENTAE 6P			
43	1	TP 0.8mm			T1
44	9	TP 1mm			T2-T8, T11-T12
45	1	VCC1-B2D			U23
46	1	XCC83008			U15
47	1	XCL8V02V044			U16
48	1	02 02 160 2201			P7
49	1	02 02 160 2201			P7
50	5	4700-008	CONDIN160P	Resistor	U9, U13, U20-U22
Total Parts Used: 211					

Página: 1

FIGURA 66 : FICHERO *.BOM

Con esta información ya se puede ir preparando la primera versión de lista de materiales, y con ello se podrá ir solicitando el material.

4.4. PLANOS

INDRA SISTEMAS dispone de departamentos en el Área de Diseño Hardware que se ocupa de llevar a cabo la delineación y ruteado de las PCBs con alternativa de subcontratación. Para el ruteado de la PCB soporte de la TIC, objeto de este Proyecto Fin de Carrera, se hace llegar a dicho departamento la versión compilada del esquemático (CDB) que contiene la información del diseño y sirve de enlace entre Design Capture y Expedition PCB.

Una vez distribuidos los componentes por la PCB sobre el plano de delineación (ver figura 10 en el punto 3.1) se observa que colocar toda la circuitería lleva al límite todos estos requisitos detallados en el punto 3.2.3, por lo que surge la necesidad de replantearse si la elección de los materiales podría ser otra para sacar más partido al espacio del que se dispone.

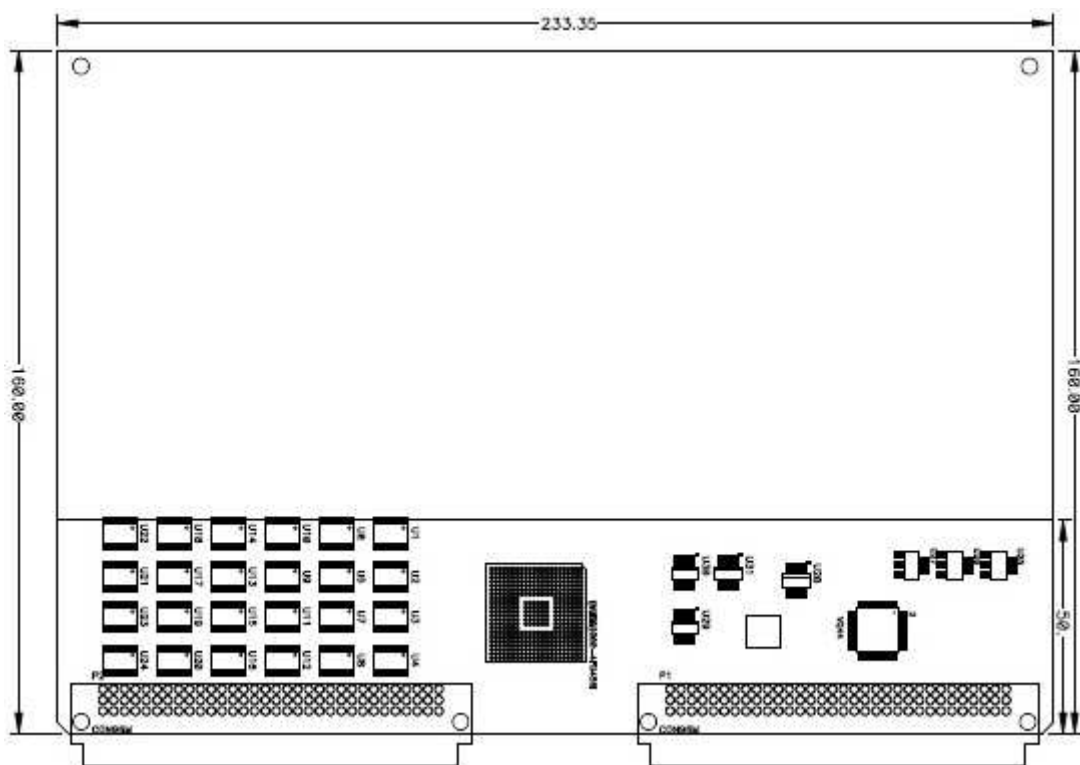


FIGURA 67 : PLANO DELINEACIÓN

Se decide que una manera de aprovechar parte del espacio es condensar todo el interfaz de señales con la tarjeta madre del EAE, en un solo conector y no en dos como se había planteado en las especificaciones. Se había ideado colocar dos conectores DIN de 96 pines, pero si se realiza la distribución de las señales sobran pines libres en ellos. Una de las ventajas de los requisitos consistía en que la elección de los conectores así como la distribución de las señales en ellos quedaba a criterio del diseñador.

El total de número de líneas requeridas es 94 líneas de señales (Tabla 13) y 7 para líneas de alimentación (Tabla 14). Con un solo conector DIN de 96 pines sería insuficiente. Pero además, se ha de tener en cuenta que conviene repetir las señales de alimentación de tantos pines como sea necesario según la corriente máxima que soporta cada uno de los pines de estos conectores DIN.

En el apartado 3.2.2.7 se realizó un análisis de consumos por alimentaciones. Éste mismo servirá para determinar el número de pines por cada señal de alimentación. Según el fabricante, cada uno de los pines del conector soporta al menos 1A. En estos casos siempre debe sobredimensionar los valores del análisis de consumos para prevenir posibles picos de consumo, y por ello se determina lo siguiente:

TENSIÓN TARJETA MADRE	Nº PINES
+15V	3
-15V	3
+12V	1
5V5D	3
3V3D	1

Además se estima que son necesarios tantos pines de masa como los pines de alimentación para no tener problemas con los retornos.

Una vez realizado el recuento de señales con la tarjeta madre, éstos son 120 pines en total lo que supone que se pueden recopilar todas ellas en un solo conector DIN41612 de 160 pines. La ocupación del mismo de un 75% de su total de 160 pines.

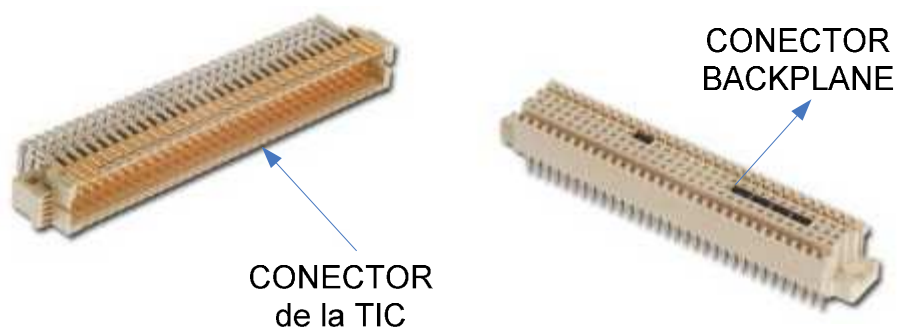


FIGURA 68 : CONECTOR DIN 41612 160 PINES

De acuerdo a esto, es necesario realizar una nueva captura de los esquemas eléctricos del diseño para incluir el nuevo conector, y a partir de ahí repetir el proceso para obtener el nuevo layout.

El nuevo plano de delineación será el presentado en la figura 69, y permitirá mantener todas las consideraciones para el diseño correcto del PCB.

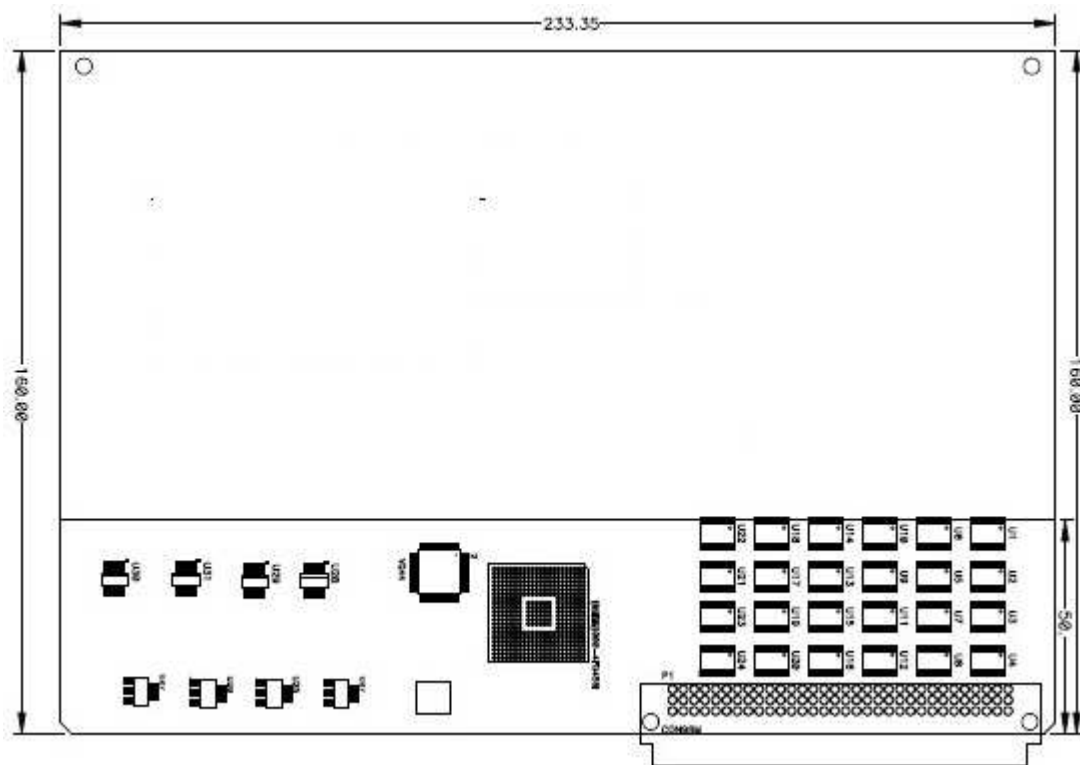


FIGURA 69 : PLANO DELINEACIÓN DEFINITIVO

Con el nuevo plano se vuelve a llevar a cabo el ruteado y se obtiene el siguiente fichero “.pcb” de Expedition PCB del paquete de software de Mentor Graphics:

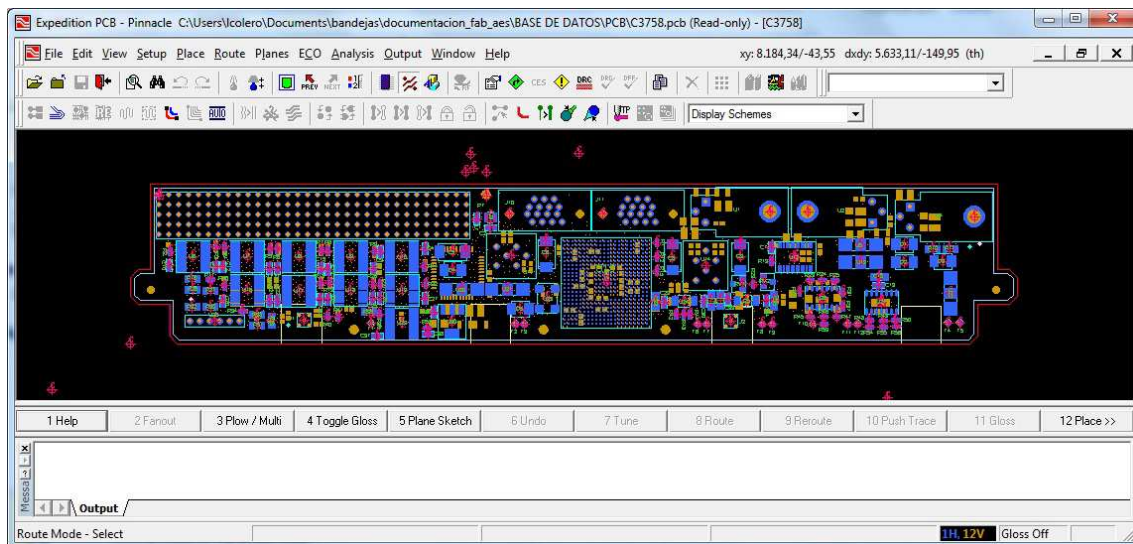


FIGURA 70 : VENTANA EXPEDITION PCB

4.5. FABRICACIÓN

INDRA SISTEMAS dispone de Departamentos de Producción para prototipos y series de grandes tiradas, pero siempre existe la alternativa de subcontratación de fabricación de ciertos módulos, como es el caso de módulos que contienen FPGAs, CPLDs o chips de alta densidad de pines.

5. DISEÑO DE LA LÓGICA PROGRAMABLE

A lo largo de este apartado se describe el diseño de la FPGA que se integra en la TIC del EAE, objeto de este Proyecto Fin de Carrera, según las especificaciones descritas.

En primer lugar, se realizará el listado de las entradas y salidas de la FPGA que se han dejado ruteadas en la PCB para su posterior uso y que define la interfaz de dicho dispositivo. Seguidamente, se procederá al estudio de la operativa de cada bloque funcional que se describió en el apartado 3 de esta memoria, con el fin de describir el circuito sintetizado en la FPGA. Se comentará en primera instancia el funcionamiento como cajas negras y se describirán los requisitos de cada bloque.

A continuación, se comentará cómo es el flujo de diseño partiendo de la creación de un proyecto hasta sintetizar el código y poder programarlo en la FPGA de la TIC.

Al final de este punto, se habrán implementado los ficheros fuente así como el resto de ficheros necesarios para programar la FPGA y que la TIC del EAE, objeto de este PFC, adquiera el comportamiento detallado en los requisitos.

5.1. ENTRADAS/SALIDAS DE LA FPGA.

Antes de comenzar a diseñar el código VHDL, es necesario definir la interfaz de la FPGA (entradas y salidas) objeto de este Proyecto Fin de Carrera.

Para ello a partir de las conexiones establecidas en la placa de circuito impreso se realiza una tabla que detalla el interfaz de entradas y salidas de la FPGA:

NOMBRE SEÑAL	DIRECCIÓN	PIN de la FPGA
FREQ_BF_15	IN	G3
FREQ_BF_14	IN	G4
FREQ_BF_13	IN	G5
FREQ_BF_12	IN	F1
FREQ_BF_11	IN	F2
FREQ_BF_10	IN	F3
FREQ_BF_9	IN	F4
FREQ_BF_8	IN	F5
FREQ_BF_7	IN	E1
FREQ_BF_6	IN	E3
FREQ_BF_5	IN	E2
FREQ_BF_4	IN	D1
FREQ_BF_3	IN	D2
FREQ_BF_2	IN	C1
FREQ_BF_1	IN	C2
FREQ_BF_0	IN	D3
RX_XILINX	IN	U21
TX_XILINX	OUT	U20
AT_BF_11	OUT	V6
AT_BF_10	OUT	W6
AT_BF_9	OUT	AB7
AT_BF_8	OUT	AA7
AT_BF_7	OUT	Y7
AT_BF_6	OUT	V7
AT_BF_5	OUT	W7
AT_BF_4	OUT	AB8
AT_BF_3	OUT	AA8
AT_BF_2	OUT	Y8
AT_BF_1	OUT	V8
AT_BF_0	OUT	W8
BLANKING_BF	OUT	V3
AT_FI1_0	OUT	AB17
AT_FI1_1	OUT	AA16
AT_FI1_2	OUT	Y16
AT_FI1_3	OUT	W16
AT_FI2_0	OUT	V16

NOMBRE SEÑAL	DIRECCIÓN	PIN de la FPGA
AT_FI2_1	OUT	AA17
AT_FI2_2	OUT	Y17
AT_FI2_3	OUT	AB18
AT_ECU1_0	OUT	V17
AT_ECU1_1	OUT	AA18
AT_ECU1_2	OUT	Y18
AT_ECU1_3	OUT	W18
AT_ECU2_0	OUT	AB19
AT_ECU2_1	OUT	AA19
AT_ECU2_2	OUT	Y19
AT_ECU2_3	OUT	AB21
BITE_CANAL_BF_1	IN	AA3
BITE_CANAL_BF_2	IN	AB3
OVERLOAD_BF_1	OUT	AB4
OVERLOAD_BF_2	OUT	AA5
AT_RF_0	OUT	V9
AT_RF_1	OUT	U9
AT_RF_2	OUT	AA10
AT_RF_3	OUT	W10
AT_RF_4	OUT	Y10
AT_RF_5	OUT	V10
AT_RF_6	OUT	U10
AT_RF_7	OUT	U11
CLK_DAC	OUT	J1
FREQ_SINT_BF_20	OUT	N1
FREQ_SINT_BF_19	OUT	T5
FREQ_SINT_BF_18	OUT	T4
FREQ_SINT_BF_17	OUT	T3
FREQ_SINT_BF_16	OUT	T2
FREQ_SINT_BF_15	OUT	R5
FREQ_SINT_BF_14	OUT	R4
FREQ_SINT_BF_13	OUT	R3
FREQ_SINT_BF_12	OUT	R2
FREQ_SINT_BF_11	OUT	P6
FREQ_SINT_BF_10	OUT	P5
FREQ_SINT_BF_9	OUT	P4
FREQ_SINT_BF_8	OUT	P3
FREQ_SINT_BF_7	OUT	R1
FREQ_SINT_BF_6	OUT	P2
FREQ_SINT_BF_5	OUT	P1
FREQ_SINT_BF_4	OUT	N6
FREQ_SINT_BF_3	OUT	N5
FREQ_SINT_BF_2	OUT	N4
FREQ_SINT_BF_1	OUT	N3
FREQ_SINT_BF_0	OUT	N2

NOMBRE SEÑAL	DIRECCIÓN	PIN de la FPGA
STROBE_SINT_BF	OUT	T1
BITE_SINT_BF	IN	AB6
FREQ_CAL_BF_14	OUT	B9
FREQ_CAL_BF_13	OUT	E10
FREQ_CAL_BF_12	OUT	C9
FREQ_CAL_BF_11	OUT	D9
FREQ_CAL_BF_10	OUT	F9
FREQ_CAL_BF_9	OUT	E9
FREQ_CAL_BF_8	OUT	A8
FREQ_CAL_BF_7	OUT	B8
FREQ_CAL_BF_6	OUT	C8
FREQ_CAL_BF_5	OUT	D8
FREQ_CAL_BF_4	OUT	A7
FREQ_CAL_BF_3	OUT	B7
FREQ_CAL_BF_2	OUT	C7
FREQ_CAL_BF_1	OUT	D7
FREQ_CAL_BF_0	OUT	E8
STROBE_CAL_BF	OUT	A9
BITE_CAL_BF	IN	AB5
SWITCH_CAL_BF_0	OUT	V2
SWITCH_CAL_BF_1	OUT	W2
FREQ_YIG_BF_11	OUT	B6
FREQ_YIG_BF_10	OUT	C6
FREQ_YIG_BF_9	OUT	A5
FREQ_YIG_BF_8	OUT	B5
FREQ_YIG_BF_7	OUT	D6
FREQ_YIG_BF_6	OUT	B4
FREQ_YIG_BF_5	OUT	C5
FREQ_YIG_BF_4	OUT	A4
FREQ_YIG_BF_3	OUT	A3
FREQ_YIG_BF_2	OUT	B3
FREQ_YIG_BF_1	OUT	C4
FREQ_YIG_BF_0	OUT	D5
STROBE_YIG_BF	OUT	A6
BITE_UFA_BF_0	IN	W5
BITE_UFA_BF_1	IN	Y5
SWITCH_FILTRO_BF_0	OUT	V4
SWITCH_FILTRO_BF_1	OUT	Y1
SWITCH_ECM_BF_0	OUT	U3
SWITCH_ECM_BF_1	OUT	U4
SDI_POT1	OUT	M1
SDI_POT2	OUT	M2
SCE_POT1	OUT	M3
SCE_POT2	OUT	M4

NOMBRE SEÑAL	DIRECCIÓN	PIN de la FPGA
SCK_POT1	OUT	M5
SCK_POT2	OUT	M6
SWITCH_ANTENA_BF_0	OUT	V1
SWITCH_ANTENA_BF_1	OUT	W1

TABLA 29: INTERFAZ ENTRADAS/SALIDAS DE LA FPGA

5.2. ARQUITECTURA DE LA FPGA.

En la figura 19, en el apartado 3.2, se presentó un diagrama de los bloques funcionales que muestra lo referente al hardware de la TIC objeto de este PFC. En consecuencia, se introdujo así la labor que la FPGA debe desempeñar para llevar a cabo estas acciones. Después de esto, en el punto 4 se detallaron los recursos hardware que vendrán presentes en la tarjeta.

Con toda esta información, el circuito codificado en VHDL dentro de la FPGA será capaz de explotar los recursos de la PCB de la forma adecuada, así como realizar las funciones necesarias para la consecución de la operativa de la TIC del EAE. El siguiente diagrama presenta la arquitectura interna que deberá contener la FPGA.

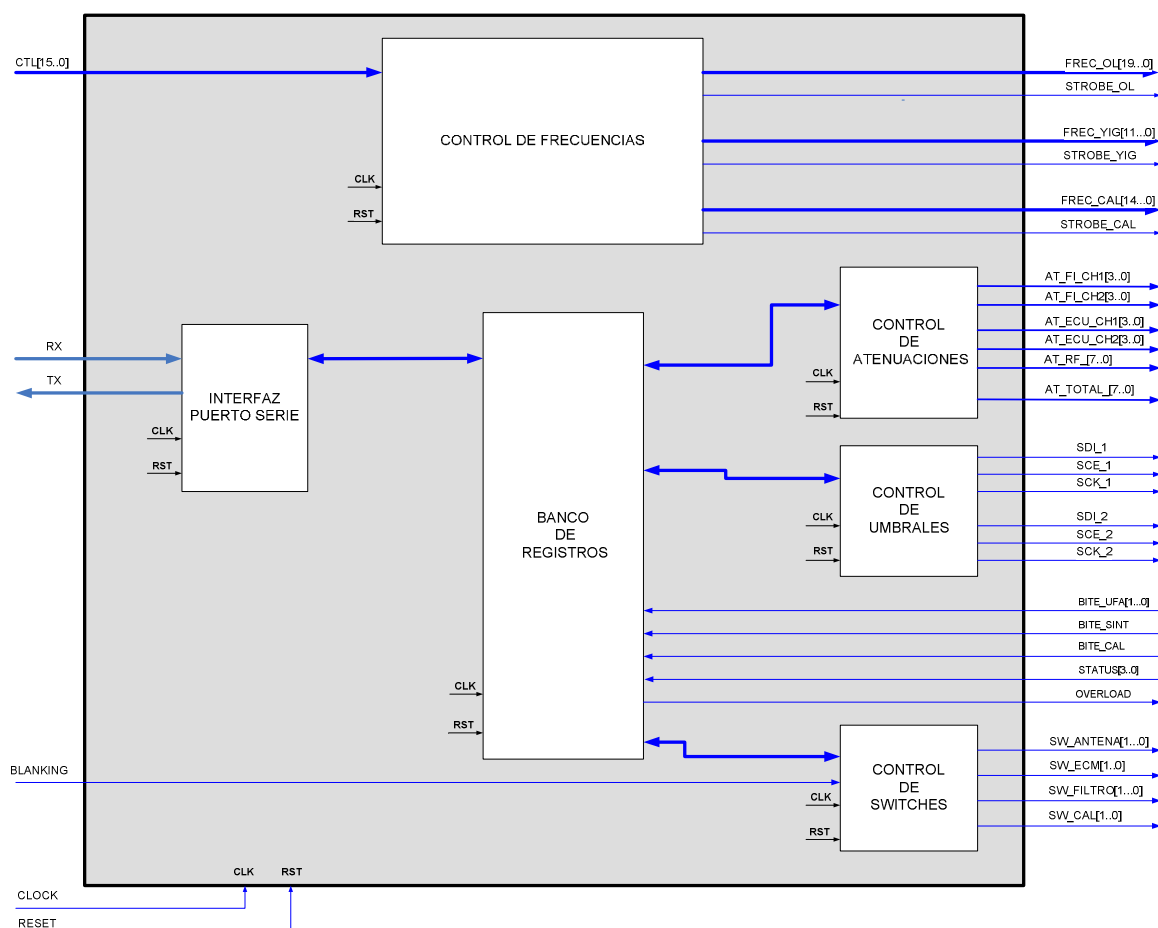


FIGURA 71 : ARQUITECTURA INTERNA DE LA FPGA

A continuación se desglosan las tareas que debe llevar a cabo la FPGA.

5.2.1. COMUNICACIONES.

La TIC del EAE, como ya se ha mencionado en apartados anteriores, se comportará de acuerdo a las órdenes que le trasmita la Tarjeta Procesadora. Para ello cuenta físicamente con las siguientes conexiones:

- Bus de 16 líneas TTL
- Puerto Serie.
- Overload.

Como especificación del sistema mediante el bus de 16 líneas se controlará la frecuencia de trabajo a la que se debe sintonizar el sistema.

Mediante el puerto serie se realizará el control del resto de tareas que coordina la TP.

Para este tipo de tarjetas interfaces donde la información debe fluir de un lado a otro, se suele implementar en la FPGA un banco de registros para disponer en diferentes momentos de la información contenida. De esta manera mediante el puerto serie, la TP podrá hacer escrituras en los registros almacenando en ellos los valores que necesita programar, así como lecturas para ver en qué estado se encuentran.

Para manejar los datos se emplearán buses de 8 bits, mientras que para manejar la dirección se empleará un bus de 16 bits. En definitiva, el acceso a la información contenida en la FPGA de la TIC se realizará basándose en la siguiente arquitectura:

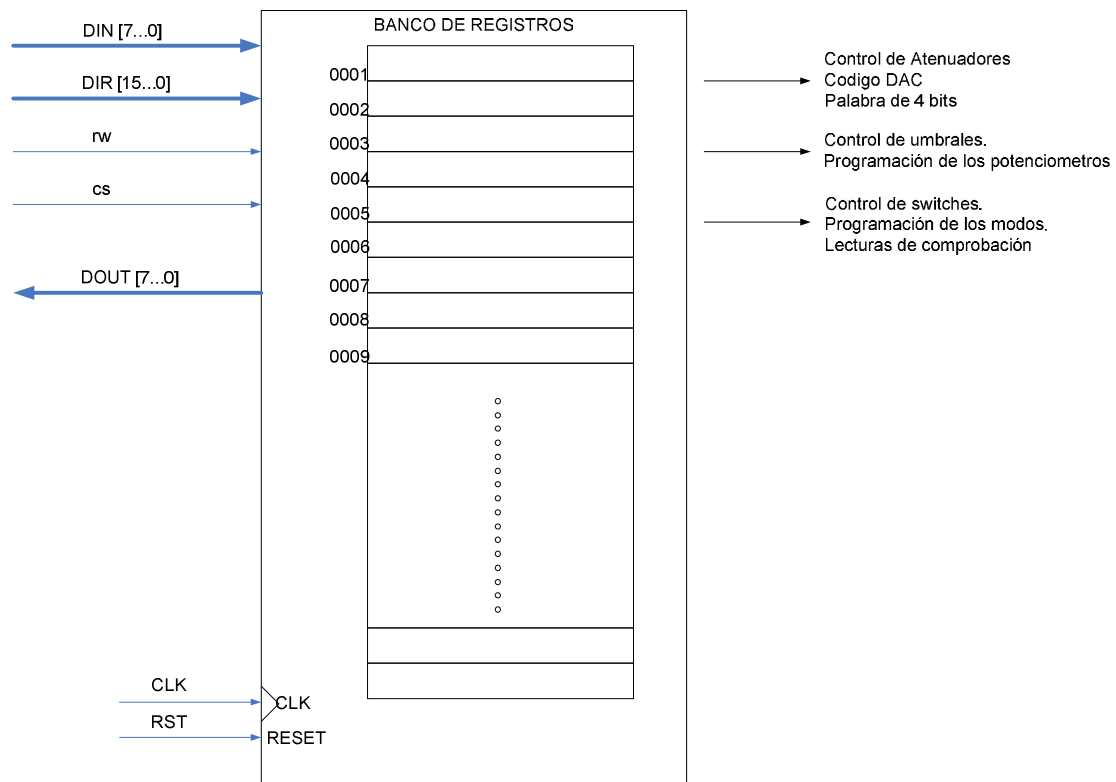


FIGURA 72 : ARQUITECTURA BANCOS DE REGISTROS

Para ello será necesario algún bloque que genere estas señales a partir de las recibidas desde el puerto serie RX y TX:

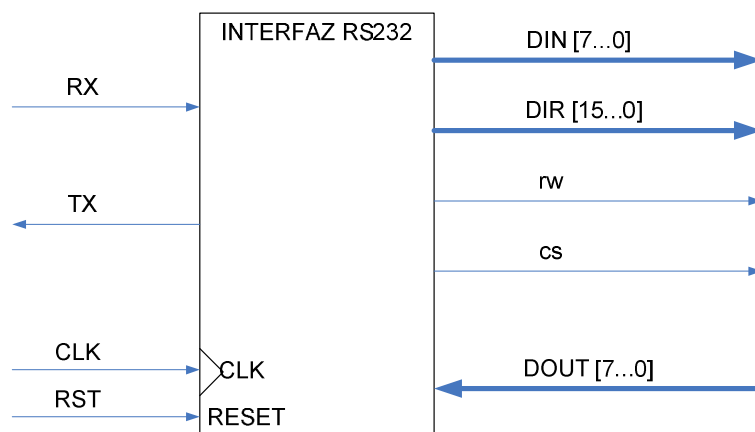


FIGURA 73 : BLOQUE INTERFAZ PUERTO SERIE

Para este caso se dispone un interfaz RS-232 ya implementado. Se trata de un módulo reutilizable de otros trabajos comunes en VHDL y considerado como una librería estándar dentro del área de ingeniería. Por lo tanto, solo se tratará de instanciarlo directamente.

5.2.2. CONTROL DE FRECUENCIAS

Según se especificaba en el estudio preliminar del diseño en el punto 3.1.1.2, se debe implementar dentro de la FPGA un bloque funcional como el siguiente:

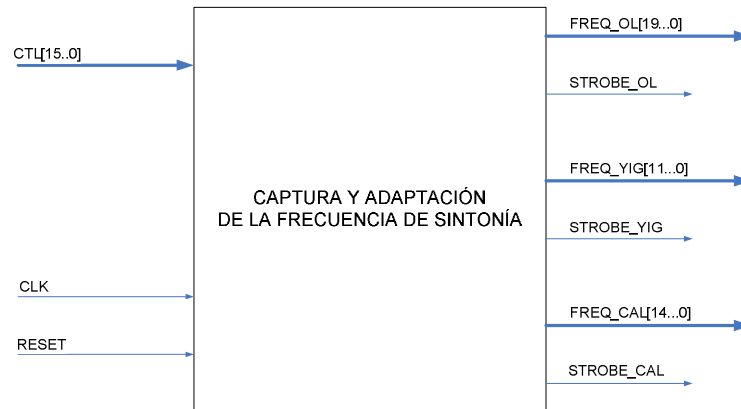


FIGURA 74 : BLOQUE CONVERSIÓN DE FRECUENCIA

Se explica a continuación de qué manera se transmite el valor de frecuencia para cada caso

5.2.2.1. SINTETIZADOR DE SINTONÍA

Según señala el fabricante, el código de frecuencia que se debe construir para que el Sintetizador de Sintonía se programe a la frecuencia marcada por la TP se basa en la siguiente ecuación:

$$OUTPUT\ FREQUENCY\ [MHz] = 2000 + (0,01 * TUNING_WORD[decimal])$$

Es decir, la FPGA de la TIC deberá ocuparse de poner la “TUNING_WORD” adecuada para que el Sintetizador se programe a la frecuencia que haya asignado la TP. Analizaremos por lo tanto la ecuación:

$$OUTPUT\ FREQUENCY\ [MHz] = 2000 + 00.1 \cdot TUNING\ WORD(decimal)$$

$$TUNING\ WORD = \frac{OUTPUT\ FREQUENCY[MHz] - 2000}{0.01}$$

$$TUNING\ WORD(decimal) = 100 * (OUTPUT\ FREQUENCY[MHz] - 2000)$$

La tabla que se puede obtener de esta ecuación es la siguiente, tal como indica el fabricante, y presentaría valores como por ejemplo los siguientes:

TUNING WORD																			
FREQUENCY [MHz]	FREQ19 (MSB)																		FREQ0 (LSB)
2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12000	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0
18000	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0

La TUNING_WORD es muestreada en flanco de bajada de un pulso que indica que hay datos válidos. Se trata de una señal denominada STROBE, que es una señal TTL y que debe tener un ancho de pulso mínimo de 100ns para se capture el dato, según indica su hoja de características.

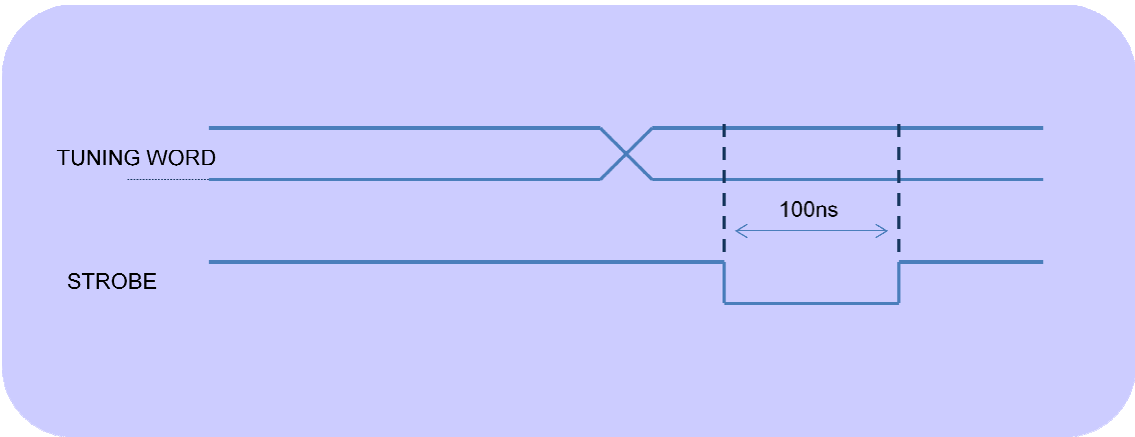


FIGURA 75 : ESPECIFICACIÓN DE SEÑAL STROBE

5.2.2.2. FILTRO YIG

De la misma manera que ocurre en el Sintetizador de Sintonía, mediante un código se programa la frecuencia de trabajo que debe adoptar el filtro YIG. La ecuación que facilita en este caso el fabricante es:

$$OUTPUT\ FREQUENCY[MHz] = 6000 + \frac{12000}{4095} * TUNING\ WORD(decimal)$$

Por lo tanto la TIC debe ocuparse de programar la siguiente palabra:

$$TUNING\ WORD(decimal) = \frac{(OUTPUT\ FREQUENCY - 6000) * 4095}{12000}$$

Igual que ocurre en los sintetizadores, se validan los datos a la entrada del filtro mediante un pulso a nivel bajo en la señal la señal STROBE_YIG

5.2.2.3. SINTETIZADOR DE CALIBRACIÓN.

El Sintetizador de Calibración funciona del mismo modo que los anteriores, de acuerdo con la información del fabricante la siguiente ecuación define la programación del sintetizador:

$$OUTPUT\ FREQUENCY[MHz] = 6000 + 0,5 * TUNING\ WORD(decimal)$$

La TIC se ocupará, igualmente, de entregar la TUNING WORD oportuna para programar la frecuencia en el Sintetizador de Calibración.

$$TUNING\ WORD(decimal) = 2 * (OUTPUT\ FREQUENCY[MHz] - 6000)$$

La tabla que se obtiene de esta ecuación es la siguiente:

TUNING WORD														
FREQUENCY [MHz]	FREQ14 (MSB)													FREQ0 (LSB)
6000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12000	0	1	0	1	1	1	0	1	1	1	0	0	0	0
19000	1	1	0	0	1	0	1	1	0	0	1	0	0	0

Se necesita igualmente una señal STROBE_CAL para indicar que las salidas son válidas y por lo tanto se debe capturar el dato de la TUNING WORD, pero en este caso será activa a nivel alto.

Por último, es en este bloque donde entra en juego el valor de desviación que se pasa a través del bus de frecuencia (ver figura 12). Las especificaciones explicaban que sería un valor de desviación entre $\pm DF$ con una resolución de B2 (MHz). Este valor además viene codificado en

complemento a 2. Teniendo en cuenta que se está operando con una resolución de $B1$. Se sabe que realiza la siguiente conversión:

FRECUENCIA DE TRABAJO (kHz)	CÓDIGO FRECUENCIA
0,00000	0000 0000 0000 0000 0000
1·B1	0000 0000 0000 0000 0001
2·B1	0000 0000 0000 0000 0010
3·B1	0000 0000 0000 0000 0011
.....
B2 = 8·B1	0000 0000 0000 1000 0000

TABLA 30 : RESOLUCIÓN DE DESVIACIÓN

La tabla de valores de desviación según el código en complemento a 2 recogido por la TIC será la siguiente:

CÓDIGO DESVIACIÓN (CTL[13...6])		VALOR DESVIACIÓN			
BINARIO	HEXADECIMAL	SIGNO	kHz	codificada (DECIMAL)	codificada (HX)
0000 0000	0	+	0	0	0
0000 0001	1	+	B2	128	80
0000 0010	2	+	2·B2	256	100
0000 0011	3	+	3·B2	384	180
0000 0100	4	+	4·B2	512	200
0000 0101	5	+	5·B2	640	280
0000 0110	6	+	6·B2	768	300
0000 0111	7	+	7·B2	896	380
0000 1000	8	+	8·B2	1024	400
...
...
0111 1101	7D	+	125·B2	16000	3E80
0111 1110	7E	+	126·B2	16128	3F00
0111 1111	7F	+	127·B2	16256	3F80
1000 0000	80	-	128·B2	16256	3F80
1000 0001	81	-	127·B2	16128	3F00
1000 0010	82	-	126·B2	16000	3E80
...
...
1111 0111	F7	-	8·B2	1024	400
1111 1000	F8	-	7·B2	896	380
1111 1001	F9	-	6·B2	768	300
1111 1010	FA	-	5·B2	640	280
1111 1011	FB	-	4·B2	512	200
1111 1100	FC	-	3·B2	384	180
1111 1101	FD	-	2·B2	256	100
1111 1110	FE	-	1·B2	128	80
1111 1111	FF	-	0	0	0

TABLA 31 : TABLA DE VALORES DE DESVIACIÓN

En resumen, para implementar el Control de Frecuencia que tiene que realizar la TIC se debe capturar el bus de datos desde la Tarjeta Procesadora en dos tandas, analizar el valor en

frecuencia que se debe programar al EAE y en consecuencia poner a las entradas de los sintetizadores y el filtro las TUNING_WORD adecuadas, así como las señales de dato válido (STROBE) que generan su carga.

5.2.3. CONTROL DE ATENUACIONES

Como se ha detallado anteriormente, una de las acciones que debe cumplir la TIC del EAE objeto de este PFC, es realizar el control de los atenuadores que están ubicados en la UMA.

La atenuación se realiza en tres etapas y será gobernada por el control de la Tarjeta Procesadora. Como se mencionó en el punto 2.1.3.1, la TP se comunicará con la TIC mediante dos líneas de puerto serie. La TIC atenderá a la orden y programará el valor correspondiente a cada atenuador.

Tanto para el valor de ecualización como el de frecuencia intermedia, sólo habrá que almacenar el valor indicado por la TP, y pasarlo a los atenuadores. La siguiente tabla presenta la relación entre la atenuación que comanda la TP, y la palabra de 4 bits codificada para cada caso:

ATENUACIÓN (solicitada por TP)	AT 3	AT 2	AT 1	AT 0
0 dB	0	0	0	0
1 dB	0	0	0	1
2 dB	0	0	1	0
3 dB	0	0	1	1
4 dB	0	1	0	0
5 dB	0	1	0	1
6 dB	0	1	1	0
7 dB	0	1	1	1
8 dB	1	0	0	0
9 dB	1	0	0	1
10 dB	1	0	1	0
11 dB	1	0	1	1
12 dB	1	1	0	0
13 dB	1	1	0	1
14 dB	1	1	1	0
15 dB	1	1	1	1

TABLA 32 : RELACIÓN VALORES DE ATENUACIÓN

Además, hay que recordar que se trata de dos canales, por lo tanto habrá dos valores para la atenuación de ecualización (ECU) y otros dos para frecuencia intermedia (FI). Los valores capturados de la TP se almacenarán en dos registros. Uno para Ecualización y otro para FI. A través de las líneas de puerto serie se comunicará estos valores y el criterio para almacenarlos en los registros es el siguiente:

REGISTRO ATENUACIÓN ECU							
CANAL 2				CANAL 1			
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AT 3	AT 2	AT 1	AT 0	AT 3	AT 2	AT 1	AT 0

REGISTRO DE ATENUACION FI							
CANAL 2				CANAL 1			
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
AT 3	AT2	AT1	AT0	AT3	AT2	AT1	AT0

El valor total de atenuación se podrá incrementar ahora en la etapa de RF en función del valor de atenuación que se reciba desde la TP. Como ya se ha explicado anteriormente, el valor de atenuación deberá ser controlado por un DAC que proporciona tensiones entre 0 y -3V. La TP se ocupa de poner el valor del código del DAC y la TIC de almacenarlo y trasladarlo al atenuador correspondiente. Esto se realiza mediante la interfaz de puerto serie, a través de la escritura del código del DAC (*código[7..0]*) el registro que corresponda:

REGISTRO ATENUACION RF							
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
<i>codigo[7]</i>	<i>codigo[6]</i>	<i>codigo[5]</i>	<i>codigo[4]</i>	<i>codigo[3]</i>	<i>codigo[2]</i>	<i>codigo[1]</i>	<i>codigo[0]</i>

Teóricamente los dos canales son iguales, esto está ya contemplado dentro de la TIC, por lo tanto con el mismo código mediante el hardware de la TIC se distribuirá el valor de tensión entre 0 y -3V a los dos atenuadores de RF.

De caracterizaciones desarrolladas en el atenuador de RF, se cuenta con una tabla que relaciona valores de atenuación a partir de su entrada en tensión. Para llegar a ese valor de tensión el DAC debe programarse según convenga.

El DAC proporciona dos salidas en corriente que serán complementarias: IOUT1 Y IOUT2. La salida IOUT1 estará cerca del valor máximo de fondo de escala cuando la salida IOUT2 se encuentre al mínimo. La relación entre las dos salidas y el código programado es la siguiente:

$$IOUT_1 = IOUT_{FS} \times \frac{CODE}{16384}$$

$$IOUT_2 = IOUT_{FS} \times \frac{16383 - CODE}{16384}$$

$$IOUT_{FS} = IOUT_1 + IOUT_2$$

Teniendo en cuenta las cargas de 50 Ω del montaje típico que indica el fabricante:

$$VOUT_1 = IOUT_1 \times R_{LOAD} = IOUT_{FS} \times \frac{CODE}{16384} \times R_{LOAD}$$

$$VOUT_2 = IOUT_2 \times R_{LOAD} = IOUT_{FS} \times \frac{16383 - CODE}{16384} \times R_{LOAD}$$

Finalmente la tabla con la que cuenta la TP para programar el atenuador de RF es la siguiente:

ATENUACION (dB)	Tensión requerida en la entrada del atenuador (V)	CODIGO DAC (HEX)
0	0	80
1	-1,32	F3
2	-1,69	15
3	-1,872	25
4	-1,983	2F
5	-2,067	36
6	-2,131	3D
7	-2,183	41
8	-2,223	45
9	-2,255	48
10	-2,283	4A
11	-2,311	4D
12	-2,331	4F
13	-2,351	51
14	-2,371	52
15	-2,391	54
16	-2,407	55
17	-2,423	57
18	-2,439	58
19	-2,455	5A
20	-2,471	5C
21	-2,485	5D
22	-2,499	5E
23	-2,515	5F
24	-2,531	61
25	-2,547	62
26	-2,567	64
27	-2,587	66
28	-2,611	68
29	-2,651	6D
29,5	-2,751	76
30	-2,92	7D

TABLA 33. RELACIÓN DE ATENUACIÓN Y CÓDIGO DEL DAC

El valor de **ATENUACION TOTAL** será compuesto por la suma del valor de atenuación de las 3 etapas y comunicado continuamente en tiempo real mediante un bus de 8 líneas a la Tarjeta Digitalizadora. Y se corresponderá a esta tabla:

Atenuación total	Valor comunicado a TD
0 dB	00
1 dB	01
2 dB	02
3 dB	03
4 dB	04
5 dB	05
6 dB	06
7 dB	07
8 dB	08
9 dB	09
10 dB	0A
11 dB	0B
12 dB	0C
13 dB	0D
14 dB	0E
15 dB	0F
16 dB	10
17 dB	11
18 dB	12
19 dB	13
20 dB	14
21 dB	15
22 dB	16
23 dB	17
24 dB	18
25 dB	19
26dB	1A
27dB	1B
28dB	1C
29dB	1D
30 dB	1E
31 dB	1F
32 dB	20
33 dB	21
34 dB	22
35 dB	23
36 dB	24
37 dB	25
38 dB	26
39 dB	27
40 dB	28
41 dB	29
42 dB	2A
43 dB	2B
44 dB	2C
45 dB	2D
60 dB	3C

TABLA 34. VALOR DE ATENUACIÓN TOTAL QUE SE INFORMARÁ A LA TARJETA DIGITALIZADORA

Por ejemplo, si se encuentran programados los atenuadores con los siguientes valores:

Atenuación RF	Atenuación FI	Atenuación ECU
10 dB	14 dB	0 dB
ATENUACION TOTAL= 10+14+0= 24dB		

La TIC le pasa a la Tarjeta Digitalizadora este total de “24dB” de esta manera:

ATENUACION TOTAL							
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	0	0	1	1	0	0	0
$24_{decimal} = 00011000_{binario} = 18_{hexadecimal}$							

Para ello igualmente, se compondrá este valor y se almacenará en un registro para que pueda tener información sobre ello la Tarjeta Procesadora, y además se mandará continuamente a la TD.

El bloque que realice esta funcionalidad será el siguiente:

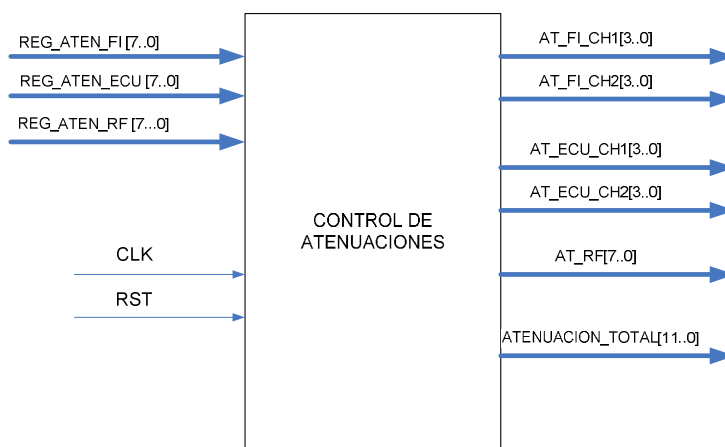


FIGURA 76: BLOQUE CONTROL DE ATENUACIONES

5.2.4. CONTROL DE UMBRALES

La FPGA debe ocuparse de generar las señales adecuadas para programar los potenciómetros digitales AD5290. Tal y como se indicó en el cronograma de la figura 29, se trata de generar las señales oportunas del interfaz SPI.

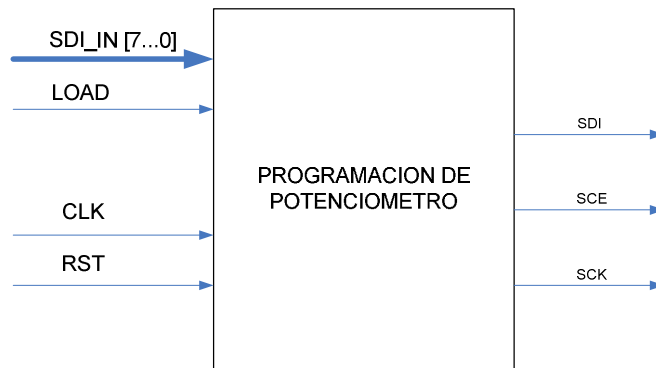


FIGURA 77 : PROGRAMACIÓN DE UN POTENCIÓMETRO SPI

La TP cargará el valor de estos umbrales al arranque y luego podrá acceder a modificarlo si lo considera oportuno. Por ello el valor SDI_IN[7...0] será capturado y almacenado en uno de los registros del banco de registros.

La TIC debe ser capaz de reconocer que debe hacer una carga sobre algún potenciómetro si detecta que se ha escrito sobre este registro un valor nuevo. En ese caso se generará una señal de LOAD que desembocará todo el proceso de programación del potenciómetro. Una vez se detecte el pulso de LOAD se considera que hay dato válido para cargar al potenciómetro y es en este momento cuando se activa la señal SCE que permanecerá en ese estado hasta que finalice la carga.

Hay que recordar que el potenciómetro digital AD5290 trabaja con una frecuencia de 4MHz mientras que el reloj del sistema es de 40MHz. Por lo tanto para que el potenciómetro funcione se le debe proporcionar una señal de reloj adecuada. Para ello se debe implementar un divisor de frecuencia.

$$f_{POTENCIOMETRO} = 4MHz$$

$$f_{SISTEMA} = 40MHz \rightarrow \frac{40}{4} = 10 \quad \text{Habrá que dividir la frecuencia al menos entre 10}$$

Para cubrir el rango se implementará un divisor por 16 utilizando 4 bits para ello ($2^4=16$). Mientras esté activa la señal *SCE* estará en marcha el divisor de frecuencia, y generará una señal cuya frecuencia es la frecuencia del sistema dividida entre 16. Esta señal será *SCK*.

Finalmente, mientras permanezca activa la señal *SCE* y cada vez que llegue un flanco de subida de *SCK*, se procederá a seleccionar un bit del valor de entrada *SDI_IN[7...0]* y se realizará comenzando por el bit de menor peso. De esa manera a través de la señal *SDI* se irá pasando el valor de la petición de la TP, de bit en bit.

5.2.5. CONTROL DE SWITCHES. MODO DE FUNCIONAMIENTO.

Desde la TP y a través de la TIC se realiza el control de los switches del EAE. Esto se realizará mediante la escritura del modo de funcionamiento por parte de la TP en un registro de la FPGA. Según este valor, la TIC a través de la FPGA debe programar los switches adecuadamente.

Según lo detallado en el punto 3.1.1.5 sobre el estudio preliminar del control de los switches del sistema, se podrá realizar un esquema como el de la siguiente figura que indica qué señales de control hay que conmutar para llevar a cabo el control sobre las señales de los switches adecuadamente:

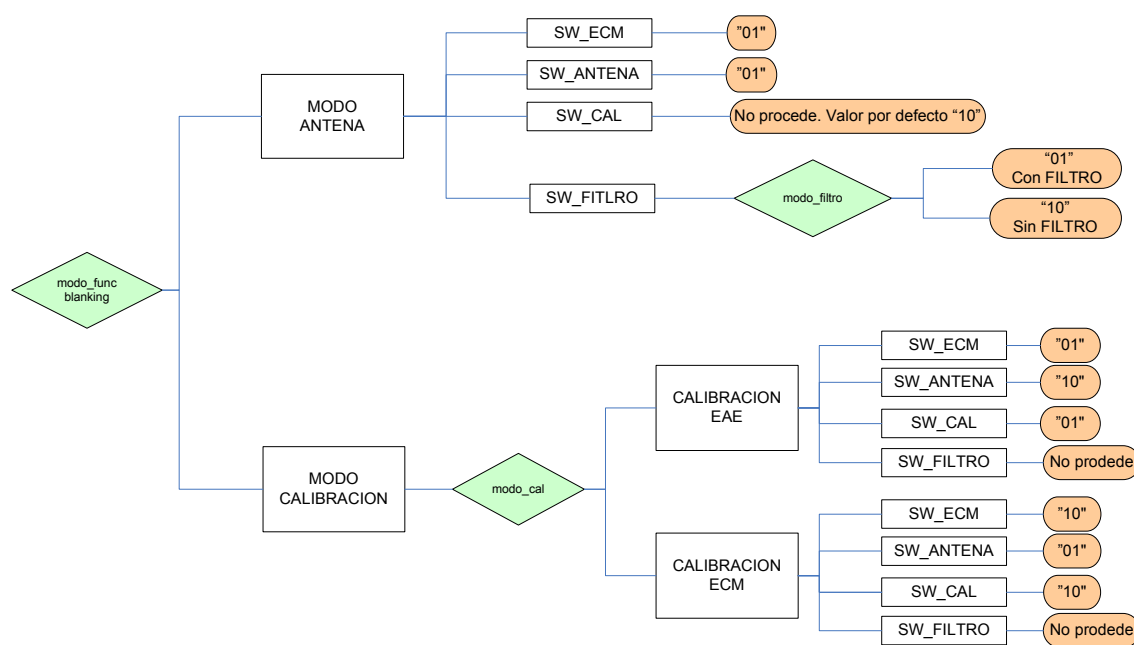


FIGURA 78 : ESQUEMA CONTROL DE MODOS DE FUNCIONAMIENTO DE LOS SWITCHES DEL EAE

De todo esto se extrae que habrá por lo tanto varios estados de las señales de estos switches atendiendo a dos señales de control: “modo de funcionamiento” y “modo de calibración”.

Según la señal “modo de funcionamiento” habrá un modo operativo normal, (MODO ANTENA), donde las señales de los switches deben adoptar los valores concretos que se especifican en el punto 3.1.1.5 y que se visualizan en el esquema anterior. El otro modo de funcionamiento será la calibración (MODO CALIBRACIÓN) y dentro de esta situación cabe destacar que entra en juego la señal “modo de calibración” que distingue entre calibrar la cadena hacia ECM o hacia el conjunto de ensamble de antenas del EAE. De la misma manera que ocurre en el MODO ANTENA, en el MODO CALIBRACIÓN las señales de los switches deben adoptar los valores especificados.

5.2.6. GESTIÓN DE SEÑALES DE AUTOCOMPROBACIÓN.

Distribuidos por las unidades del EAE se encuentran ciertos detectores de nivel cuyas señales son tomadas por la TIC como entradas a la FPGA. La TIC debe ser capaz de capturarlos y almacenarlos en los registros oportunos para que la TP tenga acceso a dicha información cuando lo requiera.

5.3. LENGUAJE VHDL.

En la descripción de requisitos se comentó que para capturar el diseño del circuito sintetizable de la FPGA, se hará empleando un lenguaje de descripción hardware, para este Proyecto Final de Carrera se utilizará el lenguaje VHDL.

El lenguaje VHDL es un lenguaje de descripción de circuitos electrónicos digitales. El significado de las siglas VHDL corresponden a VHSIC (Very High Speed Integrated Circuits) Hardware Description Language. VHDL no es un lenguaje de programación, es un lenguaje de descripción de hardware que permite describir circuitos síncronos y asíncronos. Para utilizar VHDL no se debe pensar en variables, ni funciones sino en puertas y biestables. Se debe evitar bucles y saber diferenciar qué parte del circuito es combinacional y cuál secuencial.

VHDL es un lenguaje y como tal posee sus propios tipos de datos y operadores. Los datos se almacenan en objetos que contienen valores de un tipo dado. Para conocer los identificadores y operadores que se pueden emplear se puede recurrir a cualquier manual de introducción al VHDL.

De manera resumida, la estructura básica de un archivo fuente en VHDL consiste en la elaboración de dos unidades de código VHDL:

- ✓ Declaración de Entidad: es la unidad de diseño VHDL que sirve para especificar el interfaz de los dispositivos. Una entidad es análoga a un símbolo esquemático de los diagramas electrónicos, el cual describe las conexiones del dispositivo hacia el resto del diseño.

```
29
30 --DECLARACION DE ENTIDAD
31 entity source is
32
33     Port(
34         --declaracion de entradas/salidas
35     );
36
37 end source;
38 -----
```

FIGURA 79 : ESTRUCTURA DE LA DECLARACIÓN DE ENTIDAD

- ✓ Cuerpo de Arquitectura: es la unidad de diseño VHDL que sirve para describir el funcionamiento de una “Entidad” a la que hace referencia, por lo que se puede considerar el equivalente a las tablas de verdad o a los cronogramas. Una “Arquitectura” puede tener tantos procesos como se necesite, y todos se van a

ejecutar en paralelo. Esta es la manera que tiene VHDL de expresar la concurrencia inherente al hardware.

```
40  --CUERPO DE ARQUITECTURA
41  architecture source_arch of source is
42  --declaracion de señales
43  begin
44
45  process (clk, reset,...)
46  begin
47      --codigo de descripcion
48  end process;
49
50  process (clk, reset,...)
51  begin
52      --codigo de descripcion
53  end process;
54
55  process (clk, reset,...)
56  begin
57      --codigo de descripcion
58  end process;
59
60  end source_arch;
61  -----
62
```

FIGURA 80 : ESTRUCTURA DEL CUERPO DE ARQUITECTURA

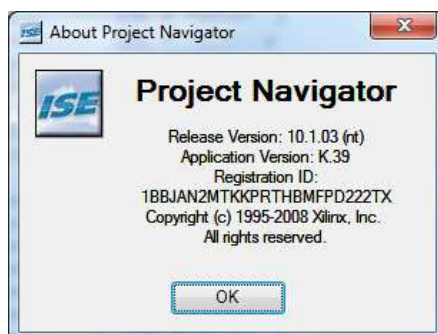
5.4. ENTORNO DE DESARROLLO.

Para el desarrollo completo de un diseño dentro de una FPGA, existen varios programas que permiten realizar esta tarea. Una de las herramientas más completas que permite realizar diseños sin necesidad de usar otra aplicación son los llamados entornos de diseño. Las herramientas de desarrollo más comunes que se encuentran en el mercado se muestran en la tabla 35:

Herramienta	Empresa	Tipo de Lenguaje
<i>Entorno ISE</i>	<i>Xilinx</i>	<i>VHDL and Esquemáticos</i>
<i>Quartus and Max+Plus</i>	<i>Altera</i>	<i>VHDL y Verilog</i>
<i>Libero IDE</i>	<i>Actel</i>	<i>VHDL y Verilog</i>

TABLA 35: HERRAMIENTAS DE DESARROLLO

En el marco de trabajo de este proyecto fin de carrera, se dispone de licencia para trabajar en el entorno de “XILINX ISE Design Suite 10.1”, porque lo que viene ya establecido tomar esta solución como herramienta software.



El ISE® (Integrated Software Environment) es el paquete de software de Xilinx® que permite desde la entrada del diseño hasta la programación del propio dispositivo Xilinx. Se puede crear, verificar, simular y sintetizar diseños basados en una FPGA. Además permite integrar otras herramientas para la simulación y verificación de los diseños como podrían ser Modelsim o Chipscope. En este Proyecto Fin de Carrera se empleará el propio simulador de “XILINX ISE Design Suite 10.1”, cuyo entorno es similar al Modelsim pero integrado en la propia herramienta.

El “ISE Project Navigator” gestiona y procesa el diseño a través de los siguientes pasos en el flujo de diseño ISE:

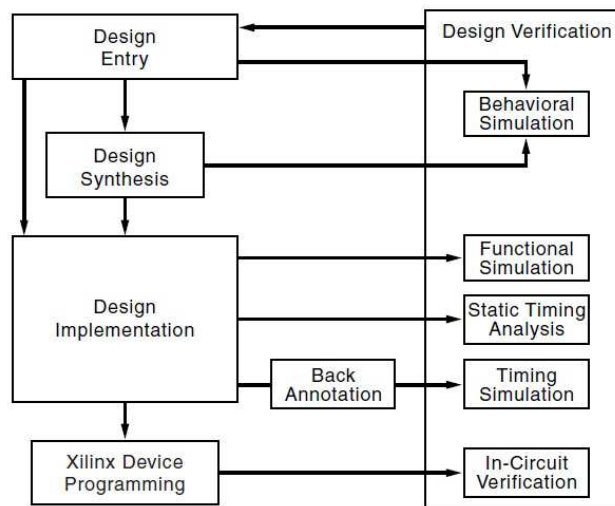


FIGURA 81 : FLUJO DE DISEÑO

- **Design Entry:** es el primer paso en el flujo de diseño. Durante este proceso, se crean los ficheros fuente sobre los que se basa el diseño. Se suele crear un fichero “*top-level*”. Se puede emplear para ello VHDL o Verilog
- **Synthesis:** es el paso que continua después de capturar el diseño mediante “Design Entry”. Este paso consiste en convertir el lenguaje del diseño en archivos de listas de conexiones (*netlist*) que son aceptados ya como entradas a la etapa de implementación.
- **Verification:** se puede verificar el funcionamiento del diseño utilizando un software para la realización de una simulación a nivel funcional. Además interpreta el código VHDL o Verilog en un circuito funcional y muestra los resultados
- **Implementation:** se permite la especificación de restricciones o indicaciones para realizar la implementación óptima sobre el dispositivo lógico programable. Se convierte el diseño de la lógica en un formato de fichero físico que puede ser cargado sobre el dispositivo. Esta herramienta incluye tres etapas principales en el diseño: Translate, Map, Place & Route.
- **Device Configuration:** Después de generar el fichero de programación, se configura el dispositivo. Durante la configuración se generan ficheros de programación y se descargarán dichos ficheros al dispositivo Xilinx desde el ordenador a través del programador JTAG (Joint Test Action Group). De este modo, es posible probar y depurar el código sobre el hardware de forma rápida y flexible permitiendo tantos cambios como sean necesarios.

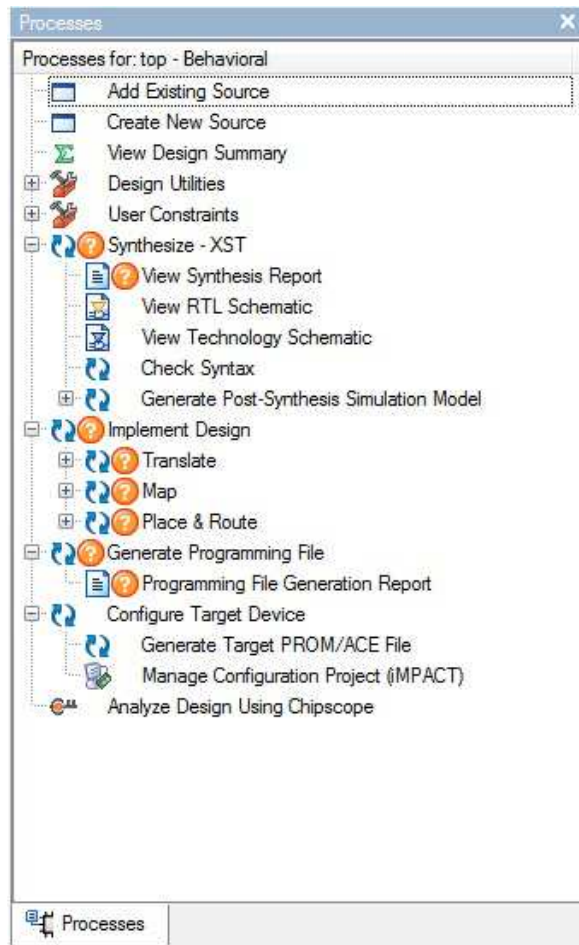


FIGURE 2: VENTANA "PROCESSES" DE ENTORNO ISE

La programación de la FPGA se realizará mediante la utilización de PROMs de Xilinx conectados a la cadena de las FPGAs. Se deben programar los PROMs utilizando ficheros creados a partir de los ficheros de configuración en la fase de diseño.

En resumen, la compilación del diseño de la FPGA genera un fichero de configuración *"*.bit"* que permite configurar la FPGA. Si la programación se realiza a través de una memoria PROM se debe generar un fichero *"*.mcs"*. Éste se consigue mediante el software iMPACT, partiendo del fichero de configuración de la FPGA *"*.bit"* se genera el fichero del formato *"*.mcs"* apto para programar memorias que contienen la configuración de la FPGA.

Por lo tanto, el software iMPACT es la herramienta de Xilinx que permite realizar las siguientes funciones: configuración de dispositivos y generación de archivos.

La configuración del dispositivo permite configurar directamente FPGAs o CPLDs de Xilinx y programar PROMs con los cables de Xilinx en varios modos. Hay cuatro cables de programación compatibles con el software iMPACT:

Name	Platforms	Voltages	Default Cable Speeds
Parallel Cable IV (Model DLC7)	PC	1.5, 1.8, 2.5, 3.3, 5V	5MHz
Platform Cable USB, Platform Cable USB II	PC	1.5, 1.8, 2.5, 3.3, 5V	6 MHz
Digilent Cables	PC	1.5, 1.8, 2.5, 3.3, 5V	10 MHz

TABLA 36: CABLES DE PROGRAMACION IMPACT

Cuando se usa este tipo de cables para configurar un dispositivo o una cadena de dispositivos, se debe establecer el modo de configuración apropiado. Se debe configurar los valores de los pines de modo: M0, M1 y M2 con el valor especificados en el datasheet del dispositivo, según el modo deseado.

En los siguientes subapartados de este punto 5.4 se describirá brevemente el desarrollo de un diseño en VHDL con ISE de Xilinx.

5.4.1. CREACIÓN DE UN PROYECTO

El área de trabajo de ISE Project Navigator presenta la siguiente forma:

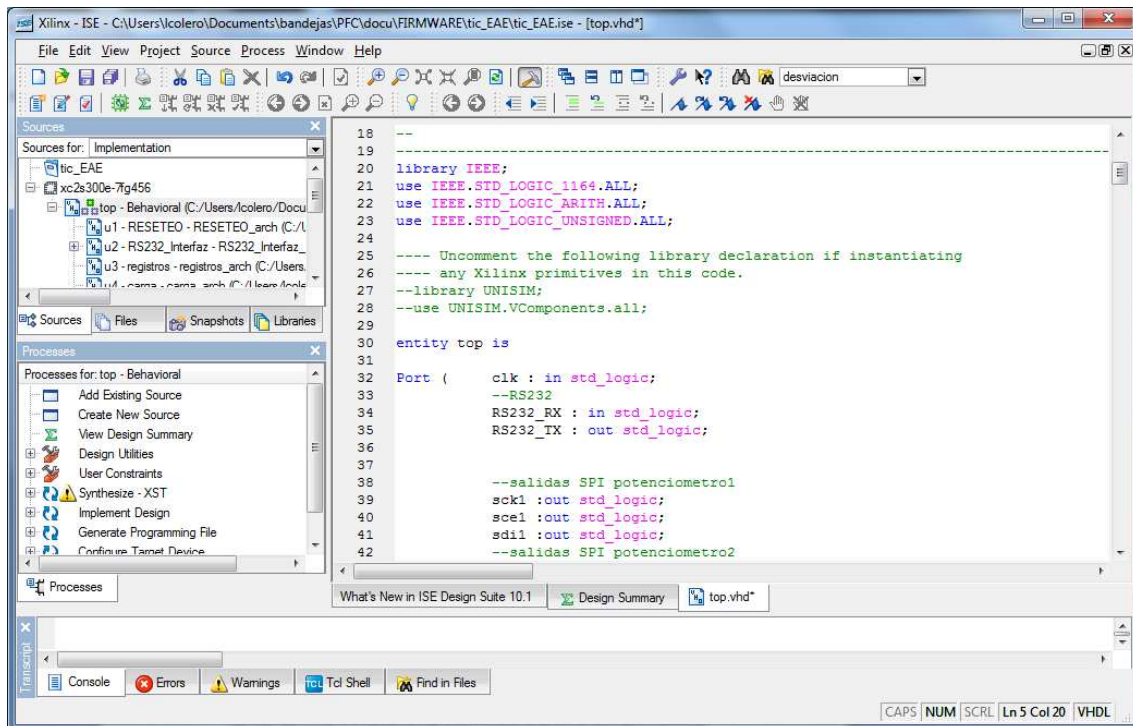


FIGURA 82: WORKSPACE DE ISE PROJECT NAVIGATOR

En primer lugar, a la hora de crear el proyecto es importante tener en cuenta dónde se crea el proyecto. La carpeta raíz llevará el nombre del proyecto. Será la carpeta principal y la de más alto nivel. Para crear un proyecto se debe ejecutar *File*→*New Project* y continuar el dialogo:

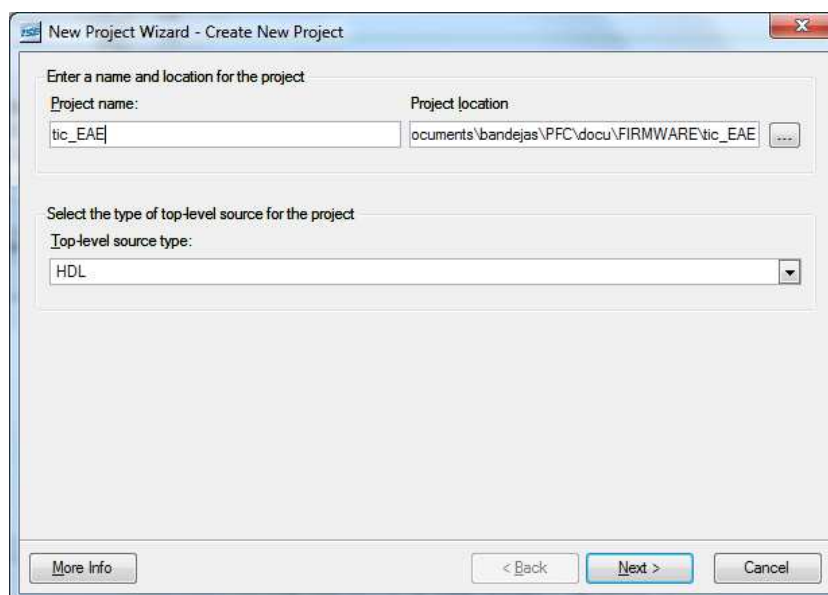


FIGURA 83 : VENTANA DIÁLOGO CREACIÓN DE UN PROYECTO

Una vez indicado el nombre y la ubicación, se seleccionará el tipo de dispositivo que se quiere programar.

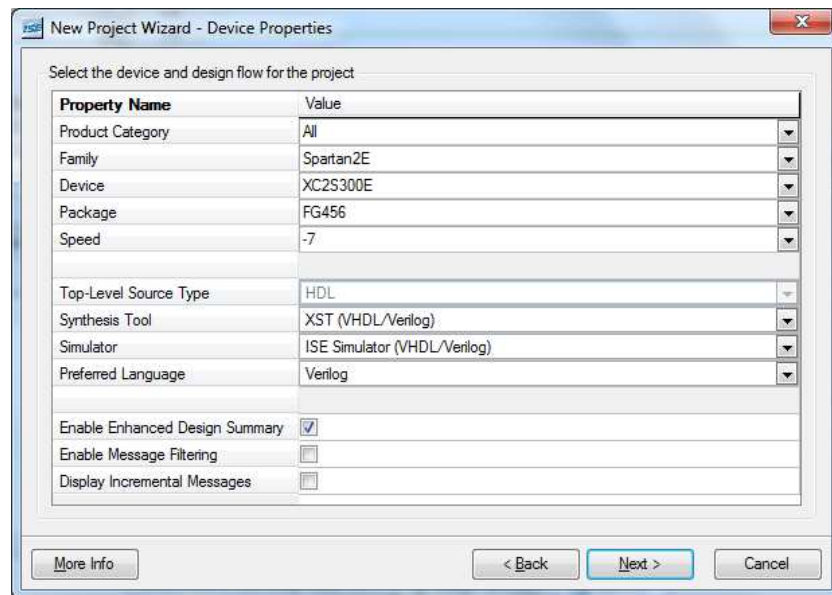


FIGURA 84 : VENTANA DEVICE PROPERTIES

En esta misma ventana de diálogo se podrá crear un fichero VHDL, que se denominará “top.vhd” y que será el fichero fuente a más alto nivel. Éste será el fichero que se sintetizará después y se volcará sobre la FPGA. Xilinx ISE posee diferentes herramientas para la entrada del diseño, se seleccionará en este diálogo que se tratará de un modulo VHDL.

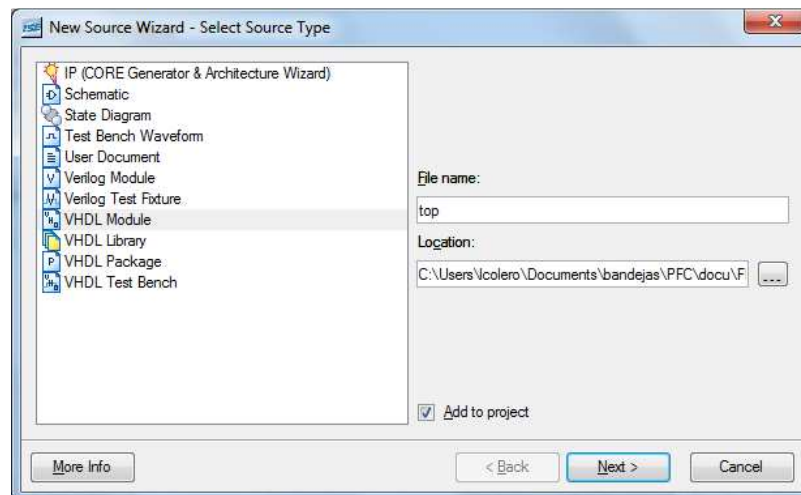


FIGURA 85 : VENTANA ELECCIÓN TIPO DE FICHERO

Una práctica habitual es crear dentro de la carpeta raíz una carpeta que se denomina “sources”. En esta carpeta se alojarán todos los archivos “*.vhd” que están asociados a cada una de las funcionalidades descritas en puntos anteriores. De esta manera quedan aquí

ordenados los ficheros fuentes y además podrán ser trasportados a cualquier otro ordenador y tan solo habrá que sintetizar.

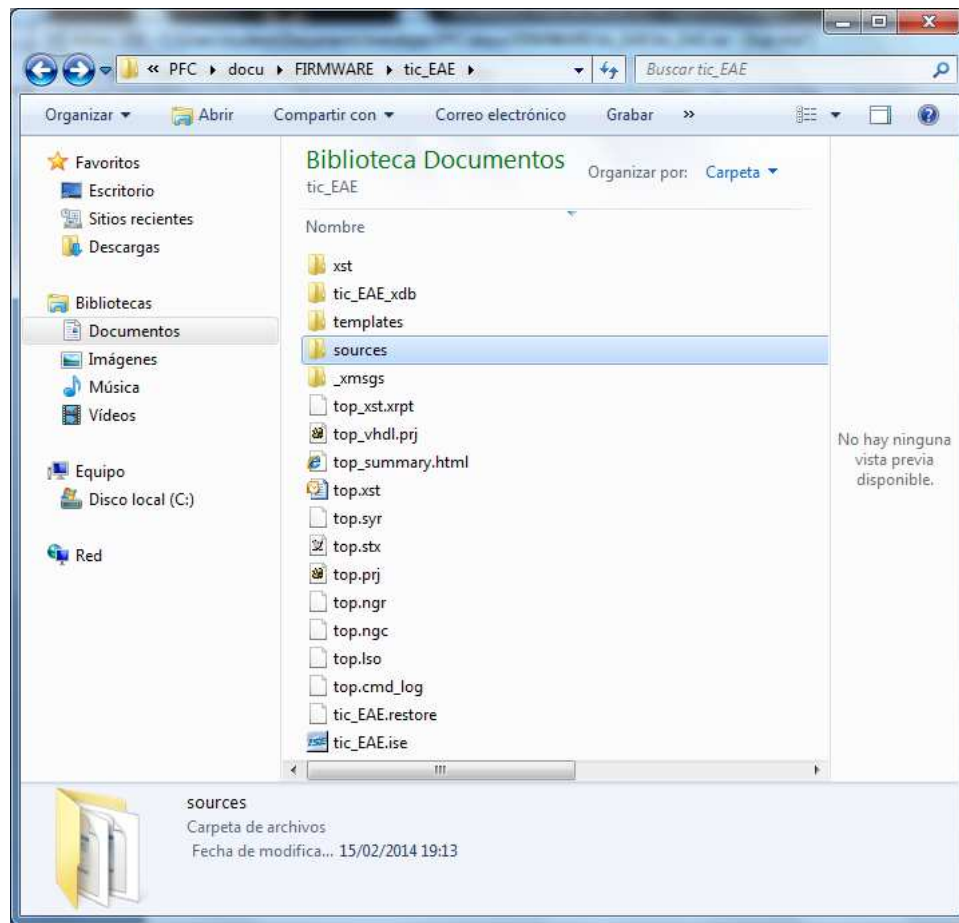


FIGURA 86 : RUTA CARPETA CON FICHEROS FUENTE

Considerando que los ficheros fuente con extensión “*.vhd” estuvieran creados en cualquier editor y guardados en dicha carpeta “sources”, se podrán agregar ahora al proyecto de acuerdo a la ventana de dialogo siguiente:

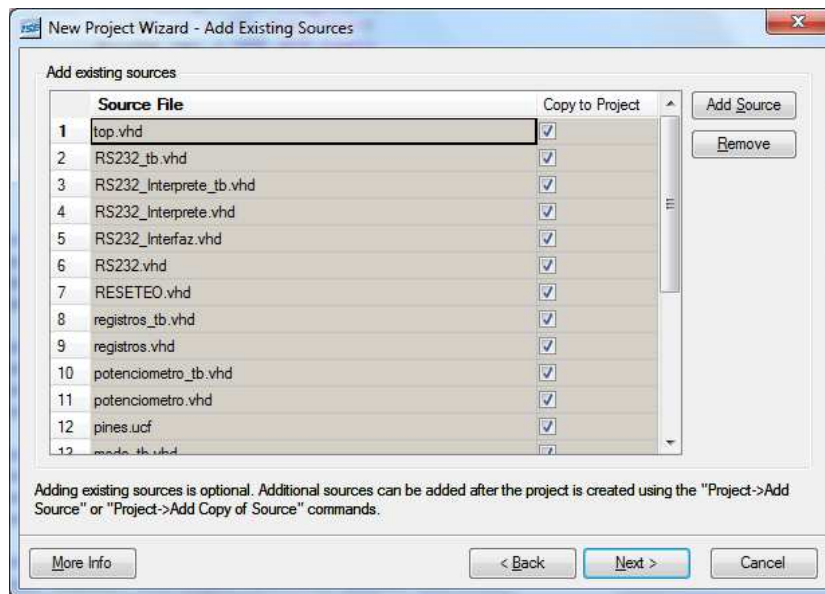


FIGURA 87 : VENTANA AÑADIR FICHEROS AL PROYECTO

Al mismo nivel de la carpeta “sources” se irán generando todos los ficheros necesarios según se vayan ejecutando los pasos en el flujo de diseño. Entre ellos se debe mencionar el fichero “top.bit” que será el fichero que se puede volcar sobre la FPGA y configura su funcionamiento.

Otro fichero importante que se ubicará en este nivel es el fichero “top.mcs”. No hay que olvidar que las FPGAs son volátiles por lo que utilizan PROMs para realizar su carga en los arranques. Este fichero “top.mcs” será volcado en la PROM y contiene la configuración de la FPGA a partir del fichero “top.bit”.

5.4.2. ENTRADA DE DISEÑO

Para la entrada de diseño cabe destacar que ISE Project Navigator cuenta con un potente conjunto de plantillas muy útiles a la hora de afrontar la codificación de un fichero fuente. Se trata de “*ISE Languages Templates*”, que se encuentra en el menú de herramientas *Edit*→*Language Templates*. Contiene una serie de construcciones y plantillas sintetizables que representan componentes lógicos de uso cotidiano como pueden ser flip-flops D, multiplexores, sumadores y diferentes tipos de primitivas. Se pueden además adaptar para añadir lo que sea oportuno para el diseño.

Una vez creados todos los ficheros fuente el aspecto del proyecto del diseño será el siguiente:

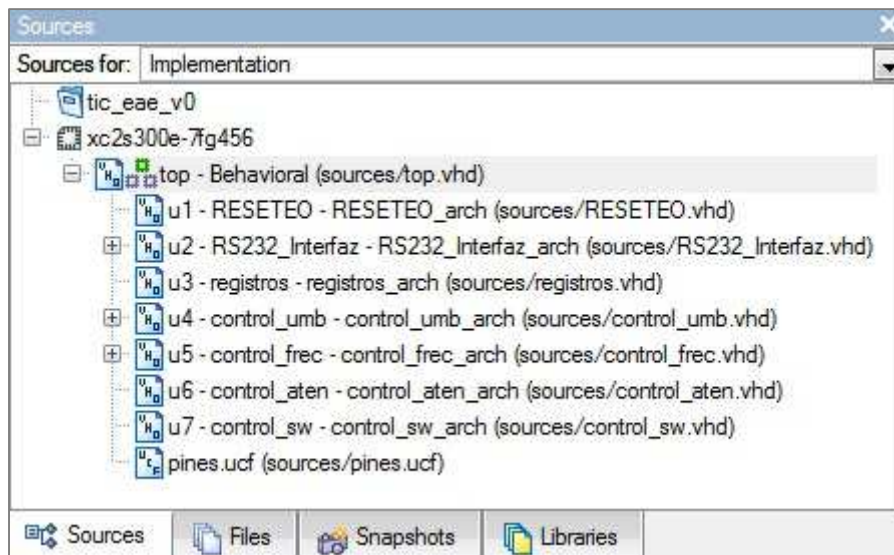


FIGURA 88 : VENTANA SOURCES

5.4.3. SÍNTESIS.

La síntesis es el proceso mediante el cual se genera una representación a nivel de puertas lógicas a partir de una descripción del comportamiento en el lenguaje del diseño, comprobando que se cumplen todas las reglas sintácticas, nombres de puertos, tipos de datos, operaciones lógicas, etc....

La síntesis se realiza sobre el fichero “*.vhd” de mayor nivel del proyecto: “top.vhd”. El proceso Synthesize-XST (Xilinx SynthesizeTechnologies) genera entre otros un fichero de extensión “top.ngc” en la misma ubicación del proyecto.

Mientras se está ejecutando la síntesis, aparecerá en la ventana inferior los mensajes correspondientes a la síntesis. Unas de las primeras acciones que se hacen en este proceso es realizar una comprobación de errores sintácticos. Si cumple con todas las reglas sintácticas se mostrará este mensaje:



FIGURA 89 : MENSAJE "CHECK SYNTAX"

Una vez acabada esta fase la ventana de procesos mostrará el siguiente aspecto:

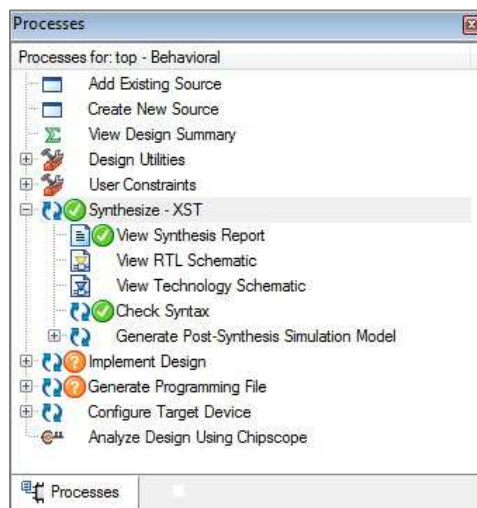


FIGURA 90 : ESTADO DE VENTANA DE PROCESOS PARA SÍNTESIS

En ocasiones junto al proceso de síntesis aparece una admiración en color amarillo. Esto significa que el proceso de síntesis ha finalizado correctamente pero con algunos avisos de tipo leve. A veces se trata de simples avisos sobre, por ejemplo, señales de buses no utilizados completamente o señales fijas a algún valor. Lo más conveniente es revisar esos avisos o bien sobre el propio diseño, o bien sobre la pagina de Xilinx para saber en lo que puede afectar al diseño.

5.4.4. ASOCIACIÓN DE PINES

Partiendo de la tabla 25 con la numeración de los pines en la FPGA, se puede proceder con la fase de asignación de pines, lo que dará lugar al fichero “pines.ucf”. Estará ubicado también dentro de la carpeta “sources”.

Este fuente es el que configura los terminales entrada/salida de la FPGA de acuerdo a lo determinado en la fase de diseño de la PCB.

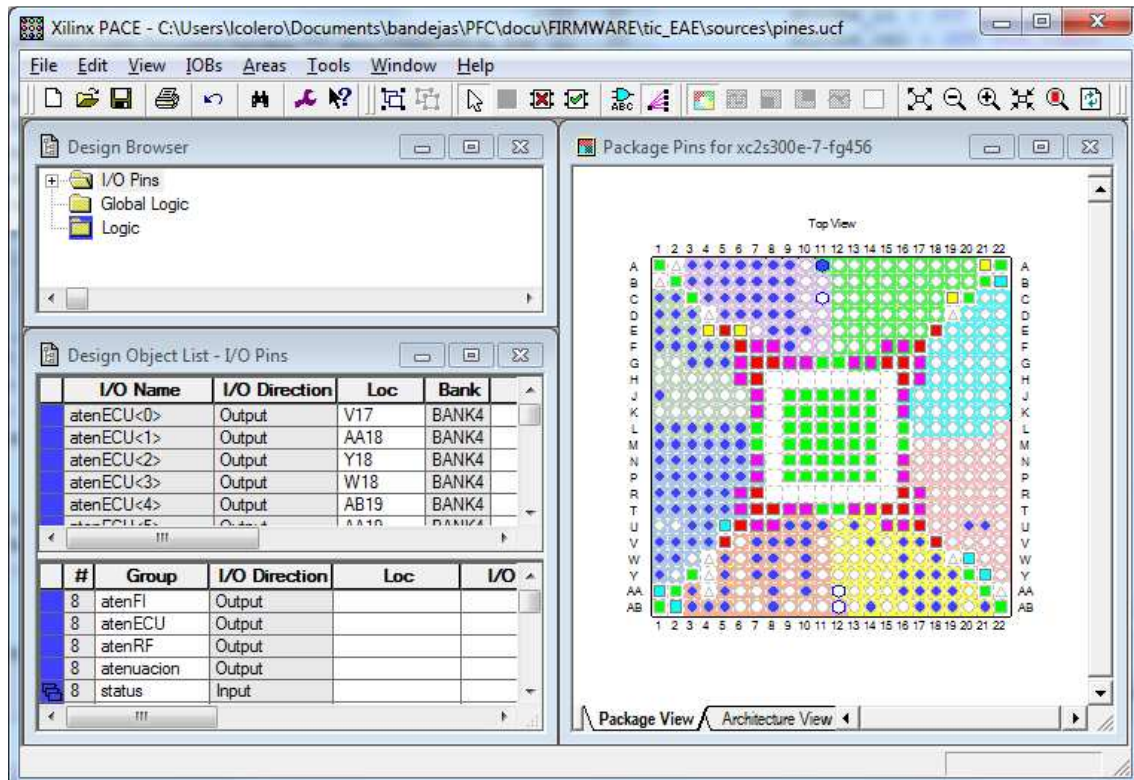


FIGURA 91 : ASOCIACIÓN DE PINES

Una vez realizada la asociación de pines, se obtiene un fichero “pines.ucf” que se podrá adaptar y modificar cuantas veces sea necesario. La siguiente figura muestra una relación entre el fichero de más alto nivel del diseño y la declaración de los pines.

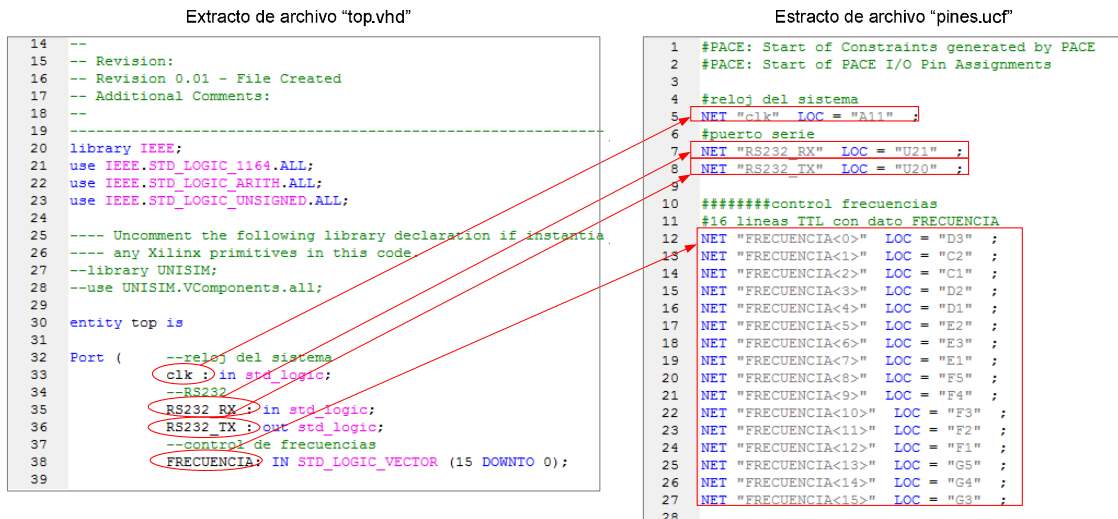


FIGURA 92 : RELACIÓN DE PINES

5.4.5. IMPLEMENTACIÓN

Tras la asignación de pines del circuito que se describía en el apartado anterior, se puede llevar a cabo ya la implementación del diseño. En este punto se introduce el diseño dentro de la estructura de la FPGA.

Las herramientas de implementación traducirán el diseño al formato adecuado, continuarán con un proceso de mapeo del diseño al dispositivo específico, y finalizarán con la colocación y ruteado en el mismo.

5.4.6. FICHERO PROGRAMACIÓN DE LA FPGA

Es en este paso donde se genera el fichero bitstream que configura la FPGA. Una vez llevados a cabo todos los pasos mencionados el aspecto de la ventana de “Processes” será el siguiente:

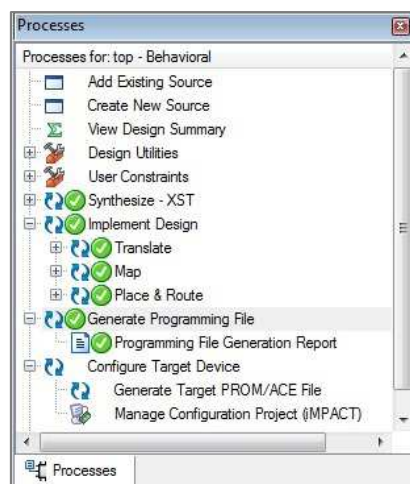


FIGURA 93 : VENTANA PROCESSES TRAS LA GENERACIÓN DEL BITSTREAM

5.4.7. FICHERO PROGRAMACIÓN DE LA PROM

Para obtener el fichero “*.mcs” que se cargará en la PROM, es necesario hacer "click" en “Configure Target Device”. Se inicia un diálogo donde se indica para qué tipo de memoria se desea generar el fichero. Una vez definida la PROM y la manera de carga, se podrá generar el fichero a partir del “top.bit”:

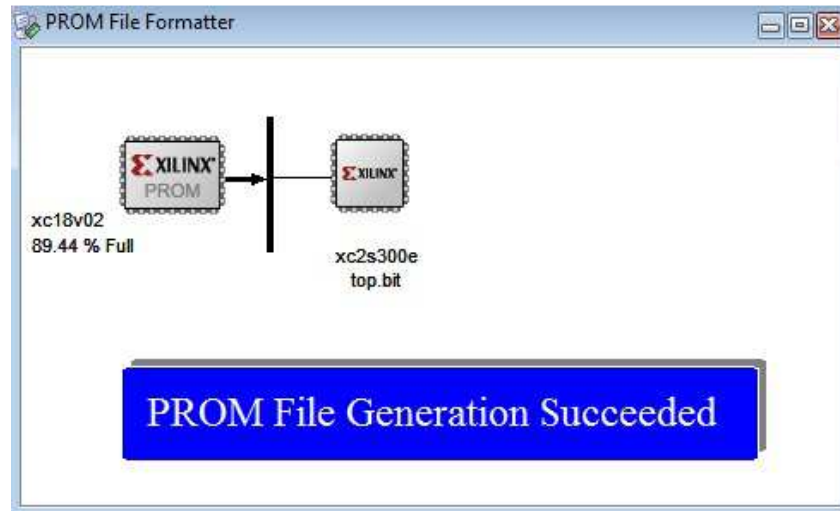


FIGURA 94 : GENERATE TARGET DEVICE

En este momento, los ficheros estarán listos para ser cargados en los dispositivos. Recuérdese que la FPGA es volátil, el fichero “top.mcs” será volcado en la PROM y contendrá la configuración de la FPGA. La cadena es la siguiente:

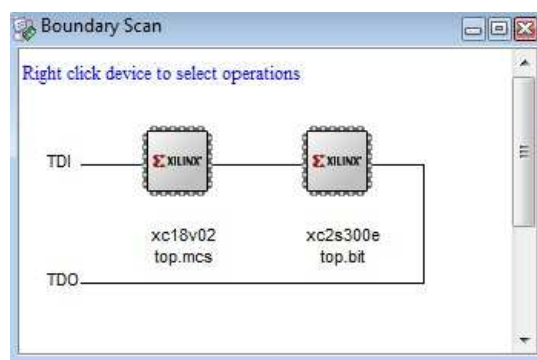


FIGURA 95 : BOUNDARY SCAN

5.5. DISEÑO DE LOS BLOQUES FUNCIONALES.

Con toda la información detallada en el punto 4 acerca del diseño HW, así como toda la información aportada en los apartados 5.1 y 5.2 se está ya en disposición de afrontar adecuadamente el diseño de la lógica programable de la FPGA, que dotará de funcionalidad a la Tarjeta Interfaz y de Control del EAE objetivo de este Proyecto Final de Carrera.

Al final de este apartado, el código estará ya implementado y se mostrará un diagrama de bloques del fichero “top-vhd” que contiene el código sintetizable. Lo que se podrá observar es una caja negra de la cual entran y salen señales. Esta caja negra no es más que el módulo VHDL del mayor nivel en la jerarquía. Éste, por lo tanto, se tratará de un diseño modular top-down que se puede dividir en diseños más sencillos de describir, para después relacionarlas a través del fichero de alto nivel. Ésta es una práctica habitual en entornos de trabajo como el de INDRA SISTEMAS S.A. debido a que permite reutilizar código de anteriores diseños y además que éstos sean más legibles y portables para diferentes proyectos.

Por tanto, el archivo “top.vhd” de mayor nivel será el propio encapsulado de la FPGA utilizando los pines adecuados numerados en la tabla 29 del punto 5.1. Por ello se observará a nivel funcional que las señales que entran y salen están ahora en relación en este circuito descrito.

En los siguientes puntos se explica cada una de las acciones que debe llevar a cabo el código de la FPGA.

5.5.1. CONTROL DE FRECUENCIAS

Según lo detallado en el punto 5.2.2 y basándose en la figura 74, la descomposición jerárquica de este bloque será el siguiente:

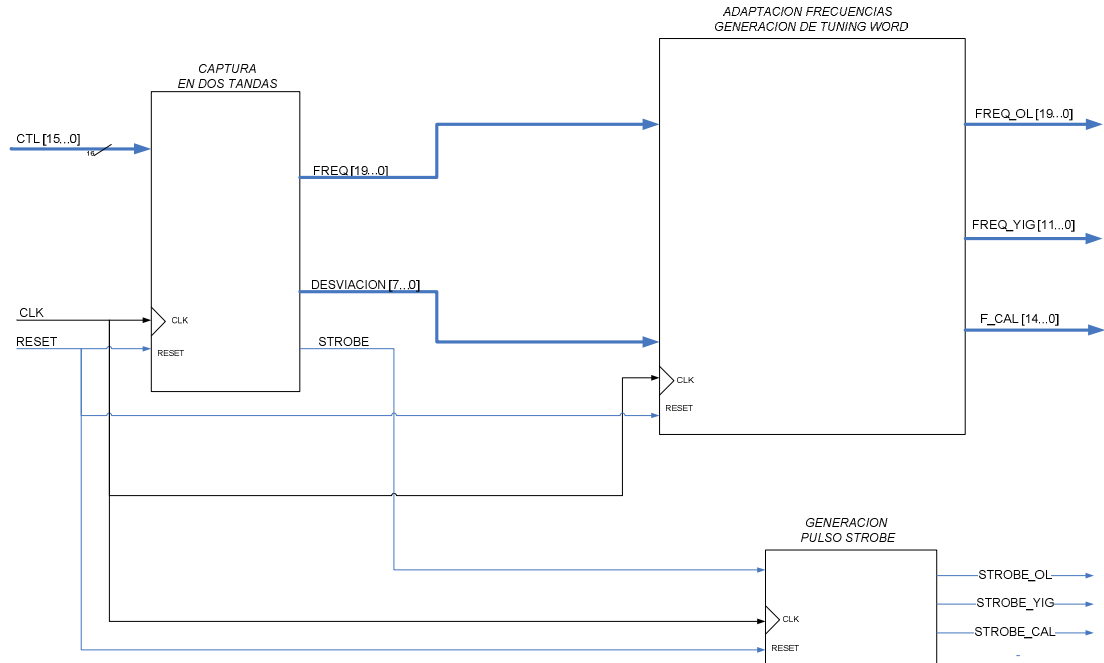


FIGURA 96 : CONTROL DE FRECUENCIAS

Una vez desarrollado el código, la declaración de entidad muestra la siguiente estructura:

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity control_freq is
31     Port (
32         clk : in std_LOGIC;
33         reset : in std_LOGIC;
34         frecuencia : in STD_LOGIC_VECTOR (15 downto 0);
35         modo: in std_LOGIC;
36         F_YIG : out STD_LOGIC_VECTOR (11 downto 0);
37         F_SINT : out STD_LOGIC_VECTOR (20 downto 0);
38         F_CAL : out STD_LOGIC_VECTOR (14 downto 0);
39         strobe_yig : OUT std_logic;
40         strobe_sint : OUT std_logic;
41         strobe_cal : OUT std_logic
42     );
43 end control_freq;

```

FIGURA 97 : DECLARACIÓN DE ENTIDAD DE MÓDULO "CONTROL_FREQ"

Y el cuerpo de arquitectura será el siguiente:

```
43
44 architecture behavioral of control_freq is
45 --señales
46 signal f_yig_bf : STD_LOGIC_VECTOR (19 downto 0);
47 signal f_sint_bf : STD_LOGIC_VECTOR (19 downto 0);
48 signal f_cal_bf : STD_LOGIC_VECTOR (19 downto 0);
49 signal strobe : std_logic;
50
51 begin
52 u1: entity work.cambia_frecuencias(cambia_frecuencias_arch)
53     port map(
54         clk      => clk,
55         reset    => reset,
56         frecuencia => frecuencia,
57         modo     => modo,
58         frec_yig  => f_yig_bf,
59         frec_sint => f_sint_bf,
60         frec_cal  => f_cal_bf,
61         strobe    => strobe
62     );
63 u2: entity work.decodifica_frecuencia(decodifica_frecuencia_arch)
64     port map(
65         clk      => clk,
66         frec_yig  => f_yig_bf,
67         frec_sint => f_sint_bf,
68         frec_cal  => f_cal_bf,
69         F_YIG     => F_YIG ,
70         F_SINT    => F_SINT ,
71         F_CAL     => F_CAL
72     );
73 u3: entity work.genStrobe(genStrobe_arch)
74     port map(
75         clk      => clk,
76         reset    => reset,
77         rg_freq  => strobe,
78         strobe_yig => strobe_yig,
79         strobe_sint=> strobe_sint,
80         strobe_cal => strobe_cal
81     );
82 end behavioral;
83
84
```

FIGURA 98 : CUERPO DE ARQUITECTURA DE "CONTROL_FREQ"

5.5.2. CONTROL DE ATENUACIONES

Según lo detallado en el punto 5.2.3 y tal como se indica en la figura 76 el diagrama de bloques que describe dicho circuito será como el siguiente:

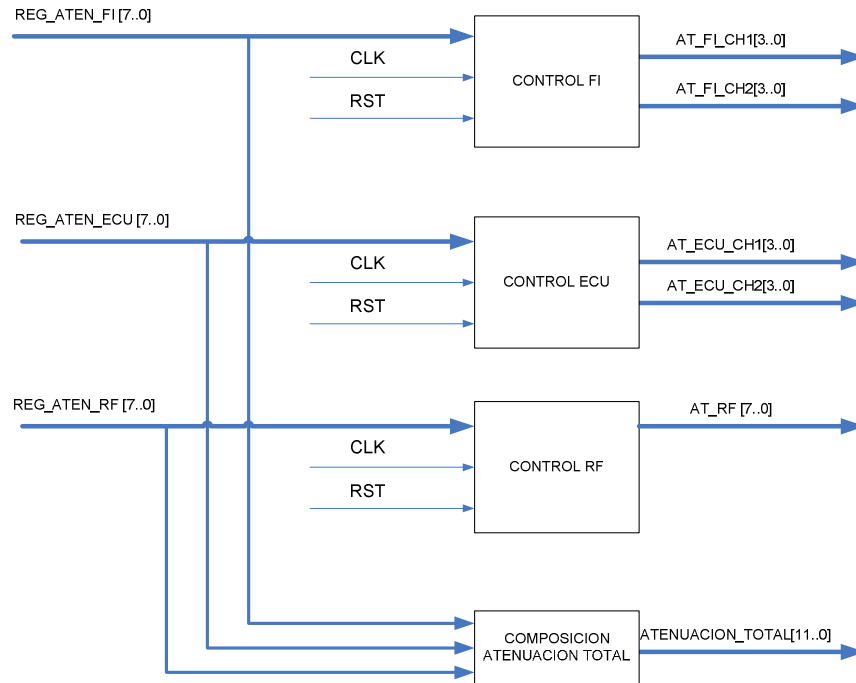


FIGURA 99: CONTROL DE ATENUACIONES

Una vez desarrollado el código, la declaración de entidad muestra la siguiente estructura:

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity control_aten is
31     Port ( clk :          in std_logic;
32           reset :        in std_logic;
33           RG_atenFI_bf: in std_logic_vector(7 downto 0);
34           RG_atenECU_bf: in std_logic_vector(7 downto 0);
35           RG_atenRF_bf: in std_logic_vector(7 downto 0);
36           atenFI_CH1:  out std_logic_vector(3 downto 0);
37           atenFI_CH2:  out std_logic_vector(3 downto 0);
38           atenECU_CH1: out std_logic_vector(3 downto 0);
39           atenECU_CH2: out std_logic_vector(3 downto 0);
40           atenRF:      out std_logic_vector(7 downto 0);
41           clk_dac:      out std_logic;
42           atenuacion_t: out std_logic_vector(7 downto 0)
43     );
44 end control_aten;
45
```

FIGURA 100 : DECLARACIÓN DE ENTIDAD DE "CONTROL_ATEN"

Y el cuerpo de arquitectura será el siguiente:

```

46 architecture control_aten_arch of control_aten is
47   --signals
48   signal atenuacion1: std_logic_vector (7 downto 0);
49   signal atenRF_bf: std_logic_vector (7 downto 0);
50   begin
51   --*FI*#####
52   --el valor de atenuacion de FI es siempre el escrito por puerto serie desde la TP
53   atenFI_CH1<=RG_atenFI_bf(7 downto 4);
54   atenFI_CH2<=RG_atenFI_bf(3 downto 0);
55   --*ECU*#####
56   --el valor de atenuacion de FI es siempre el escrito por puerto serie desde la TP
57   atenECU_CH1<=RG_atenECU_bf(7 downto 4);
58   atenECU_CH2<=RG_atenECU_bf(3 downto 0);
59   -----
60   --*RF*#####
61   --segun el codigo hx del DAC este atenuador da los siguientes valores en dB -----
62   with RG_atenRF_bf select
63     atenRF_bf <=  X"00" when X"80", --0 dB
64                   X"01" when X"F3", --1 dB
65                   X"02" when X"15", --2 dB
66                   X"03" when X"25", --3 dB
67                   X"04" when X"2F", --4 dB
68                   X"05" when X"36", --5 dB
69                   X"06" when X"3D", --6 dB
70                   X"07" when X"41", --7 dB
71                   X"08" when X"45", --8 dB
72                   X"09" when X"48", --9 dB
73                   X"0A" when X"4A", --10 dB
74                   X"0B" when X"4D", --11 dB
75                   X"0C" when X"4F", --12 dB
76                   X"0D" when X"51", --13 dB
77                   X"0E" when X"52", --14 dB
78                   X"0F" when X"54", --15 dB
79                   X"10" when X"55", --16 dB
80                   X"11" when X"57", --17 dB
81                   X"12" when X"58", --18 dB
82                   X"13" when X"5A", --19 dB
83                   X"14" when X"5C", --20 dB
84                   X"15" when X"5D", --21 dB
85                   X"16" when X"5E", --22 dB
86                   X"17" when X"5F", --23 dB
87                   X"18" when X"61", --24 dB
88                   X"19" when X"62", --25 dB
89                   X"1A" when X"64", --26 dB
90                   X"1B" when X"66", --27 dB
91                   X"1C" when X"68", --28 dB
92                   X"1D" when X"6D", --29 dB
93                   X"1E" when X"76", --29.5 dB
94                   X"00" when others;
95   atenRF<=atenRF_bf;
96   -----
97   --*ATENUACION TOTAL*#####
98   -- Informacion para la TD.
99   -- Composicion de los valores de los atenuadores, informacion para la TD.
100  -- Es la suma de los valores en dB de cada etapa
101  atenuacion1 <= (RG_atenECU_bf and X"0F") + (RG_atenFI_bf and X"0F");
102  atenuacion_t <= atenuacion1 + atenRF_bf;
103  --*reloj de dac
104  clk_dac <= clk;
105  end control_aten_arch;
106
107

```

FIGURA 101 : CUERPO DE ARQUITECTURA DE "CONTROL_ATEN"

5.5.3. CONTROL DE UMBRALES

En el punto 5.2.4 se detalla en qué consiste la carga de un valor en un potenciómetro digital con interfaz SPI. La descomposición jerárquica de este bloque de control de umbrales será:

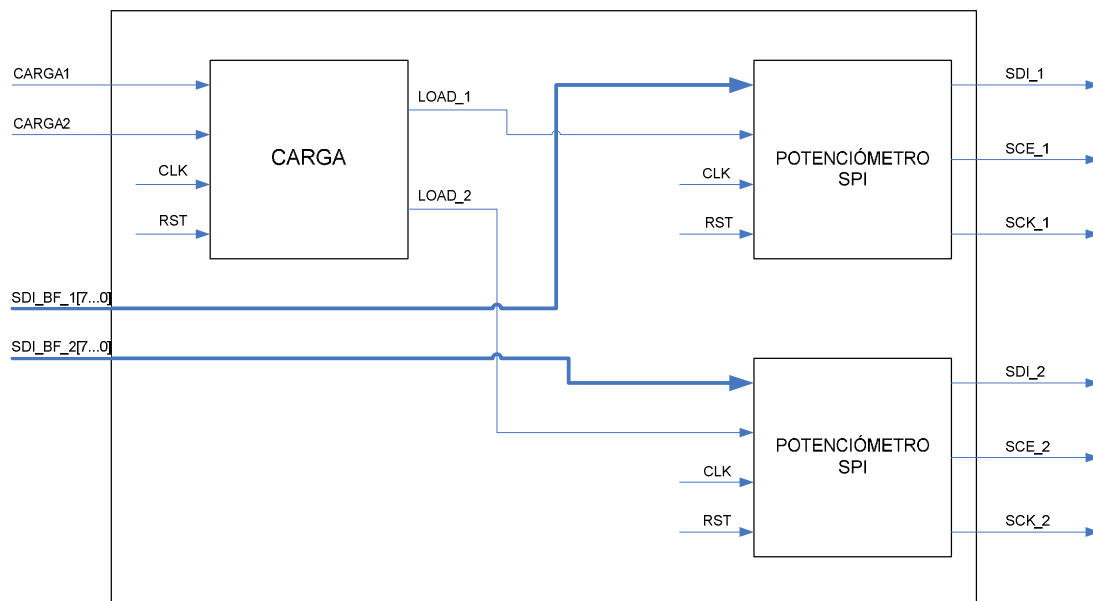


FIGURA 102 : BLOQUE CONTROL DE UMBRALES

El código desarrollado para cumplir esta función tendrá la siguiente declaración de entidad:

```

19  -----
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  ---- Uncomment the following library declaration if instantiating
26  ---- any Xilinx primitives in this code.
27  --library UNISIM;
28  --use UNISIM.VComponents.all;
29
30  entity control_umb is
31  Port ( clk : in  STD_LOGIC;
32        reset : in  STD_LOGIC;
33        sdi_bf_1 : in  STD_LOGIC_VECTOR (7 downto 0);
34        sdi_bf_2 : in  STD_LOGIC_VECTOR (7 downto 0);
35        carga1 : in  STD_LOGIC;
36        carga2 : in  STD_LOGIC;
37
38        --output SPI
39        sck1 : out  STD_LOGIC;
40        sce1 : out  STD_LOGIC;
41        sdi1 : out  STD_LOGIC;
42
43        sck2 : out  STD_LOGIC;
44        sce2 : out  STD_LOGIC;
45        sdi2 : out  STD_LOGIC
46  );
47  end control_umb;
48

```

FIGURA 103 : DECLARACIÓN DE ENTIDAD DE "CONTROL_UMB"

Y el cuerpo de arquitectura será el siguiente:

```
49 architecture Behavioral of control_umb is
50
51 signal load_pot1, load_pot2: std_logic;
52
53 begin
54 --potenciometro1
55 u1: entity work.loadPot (loadPot_arch)
56     port map(
57
58         clk          =>    clk,
59         reset        =>    reset,
60         --entradas
61         rg_pot1      =>    carga1,
62         rg_pot2      =>    carga2,
63         --salidas, señal que generará
64         load_pot1    =>    load_pot1,
65         load_pot2    =>    load_pot2
66     );
67
68 --potenciometro 2
69 u2: entity work.potenciometro (potenciometro_arch)
70     port map(
71
72         clk          =>    clk,
73         rst          =>    reset,
74         sdi_bf       =>    sdi_bf_1,
75         carga        =>    load_pot1,
76         --señales SPI
77         sck1         =>    sck1,
78         scl1         =>    scl1,
79         sdi1         =>    sdi1
80     );
81
82 --señal de carga
83 u3: entity work.potenciometro (potenciometro_arch)
84     port map(
85
86         clk          =>    clk,
87         rst          =>    reset,
88         sdi_bf       =>    sdi_bf_2,
89         carga        =>    load_pot2,
90         --señales SPI
91         sck1         =>    sck2,
92         scl1         =>    scl2,
93         sdi1         =>    sdi2
94     );
95
96
97 end Behavioral;
```

FIGURA 104 : CUERPO DE ARQUITECTURA DE "CONTROL_UMB"

5.5.4. CONTROL DE SWITCHES

Tal y como se especificó en el punto 5.2.5 y según la figura 78 se prepara la siguiente unidad de código que cumple dicha especificación.

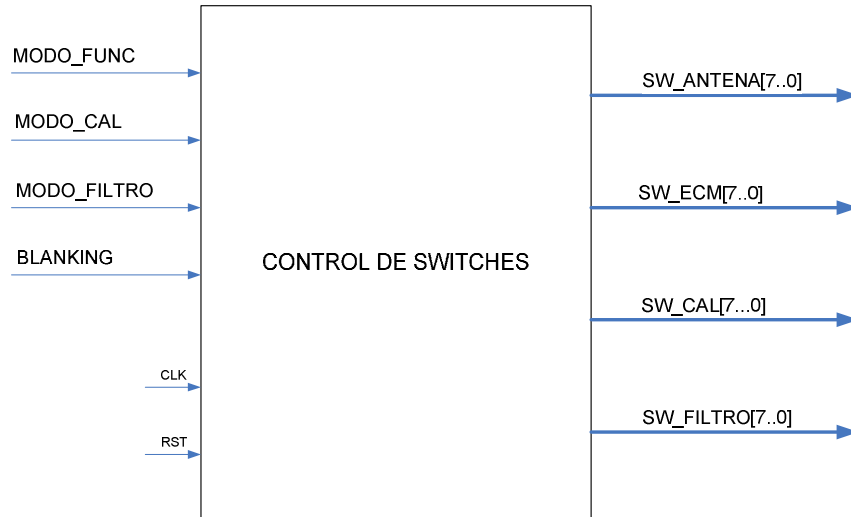


FIGURA 105: BLOQUE CONTROL DE SWITCHES

La declaración de entidad es la siguiente:

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library declaration if instantiating
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity control_sw is
31     Port ( clk : in  STD_LOGIC;
32           reset : in  STD_LOGIC;
33           modo_func : in  STD_LOGIC;
34           modo_cal : in  STD_LOGIC;
35           modo_filtro : in  STD_LOGIC;
36           sw_ufa : out  STD_LOGIC_VECTOR (7 downto 0);
37           sw_antena : out  STD_LOGIC_VECTOR (7 downto 0);
38           sw_cal : out  STD_LOGIC_VECTOR (7 downto 0);
39           sw_filtro : out  STD_LOGIC_VECTOR (7 downto 0);
40           blanking : in  STD_LOGIC
41     );
42
43 end control_sw;
```

FIGURA 106 : DECLARACIÓN DE ENTIDAD DE "CONTROL_SW"

5.5.5. BANCO DE REGISTROS

Este bloque funcional consiste en una serie de registros que contendrán los valores. Se ha preparado la siguiente tabla para identificar los registros que se implementarán en la FPGA:

DIRECCIÓN	NOMBRE	DESCRIPCIÓN	WR/RD
0000 _H	RG_version_bf [7...0]	Registro cargado con la versión de FW. Sirve para que la TP sepa que versión de FW lleva cargada la tarjeta	NO/SI
0001 _H	RG_biteCOM [7...0]	Registro habilitado para verificar la comunicación entre la TP y la TIC.	SI/SI
2001 _H	RG_potenciómetro1_bf [7...0]	Registro habilitado para cargar el valor del potenciómetro, para generar el valor umbral del detector del canal 1	SI/SI
2002 _H	RG_potenciómetro2_bf [7...0]	Registro habilitado para cargar el valor del potenciómetro, para generar el valor umbral del detector del canal 2	SI/SI
2005 _H	RG_atenuacionFI_bf [7...0]	Registro habilitado para cargar el valor de la atenuación FI.	SI/SI
2006 _H	RG_atenuacionECU_bf [7...0]	Registro habilitado para cargar el valor de la atenuación ECU.	SI/SI
2007 _H	RG_atenuacionRF_bf [7...0]	Registro habilitado para cargar el valor de la atenuación RF	SI/SI
2008 _H	RG_status_bf [7...0]	Registro que contiene el valor de las señales de autocomprobación de la UMA.	NO/SI
2009 _H	RG_atenuacion_bf [7...0]	Registro que contiene el valor total de atenuación que se debe pasar como información a la TD.	NO/SI
2013 _H	RG_biteUFA_bf [7...0]	Registro que contiene el valor de las señales de autocomprobación de la UFA.	NO/SI
2014 _H	RG_biteCAL_bf [7...0]	Registro que contiene el valor de la señal de autocomprobación de la UC.	NO/SI
2015 _H	RG_biteSINT_bf [7...0]	Registro que contiene el valor de la señal de autocomprobación de la UOL	NO/SI
2016 _H	RG_mod0_bf [7...0]	Registro habilitado para cargar el modo de funcionamiento del EAE	SI/SI
2017 _H	RG_mod0CAL_bf [7...0]	Registro habilitado para cargar el modo de calibración del EAE	SI/SI
2018 _H	RG_mod0FILTRO_bf [7...0]	Registro habilitado para habilitar o deshabilitar el FILTRO de la UFA	SI/SI
2022 _H	RG_swECM	Registro que contiene el valor programado del switch de la UFA	NO/SI
2023 _H	RG_swCAL	Registro que contiene el valor programado del switch de la UC	NO/SI
2024 _H	RG_swANTENA	Registro que contiene el valor programado del switch del conjunto de Ensamble de Antenas	NO/SI
2025 _H	RG_swFILTRO	Registro que contiene el valor programado del switch del filtro de la UFA	NO/SI

TABLA 37 : BANCO DE REGISTROS

La declaración de entidad es la siguiente:

```

11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.STD_LOGIC_ARITH.ALL;
14 use IEEE.STD_LOGIC_UNSIGNED.ALL;
15
16 -- Uncomment the following lines to use the declarations that are
17 -- provided for instantiating Xilinx primitive components.
18 -- library UNISIM;
19 -- use UNISIM.VComponents.all;
20
21 entity registros is
22     Port ( clk : in std_logic;
23           reset : in std_logic;
24           --puerto serie
25           MIF_dout : out std_logic_vector(7 downto 0);
26           MIF_din : in std_logic_vector(7 downto 0);
27           MIF_dir : in std_logic_vector(15 downto 0);
28           MIF_rw : in std_logic;
29           MIF_cs : in std_logic;
30
31           -----
32           --valor cargado para cada potenciómetro
33           RG_pot1_bf: out std_logic_vector(7 downto 0);
34           RG_pot2_bf: out std_logic_vector(7 downto 0);
35           -----
36           --valores cargados de atenuadores
37           RG_atenFI_bf: out std_logic_vector(7 downto 0);
38           RG_atenECU_bf: out std_logic_vector(7 downto 0);
39           RG_atenRF_bf: out std_logic_vector(7 downto 0);
40           RG_status_in: in std_logic_vector(7 downto 0);
41           RG_atenuacion_bf: out std_logic_vector(7 downto 0);
42           -----
43           --señales de autocomprobación
44           RG_biteUFA_in: in std_logic_vector(7 downto 0);
45           RG_biteCAL_in: in std_logic_vector(7 downto 0);
46           RG_biteSINT_in: in std_logic_vector(7 downto 0);
47           --señales de los switches
48           RG_swUFA_in: in std_logic_vector(7 downto 0);
49           RG_swCAL_in: in std_logic_vector(7 downto 0);
50           RG_swANTENA_in: in std_logic_vector(7 downto 0);
51           RG_swFILTRO_in: in std_logic_vector(7 downto 0);
52           -----
53           --origen de carga-----ocurre cuando escribimos en el registro
54           RG_pot1_out: out std_logic;
55           RG_pot2_out: out std_logic;
56           -----
57           --modos de funcionamiento
58           RG_modobf: out std_logic_vector(7 downto 0);
59           RG_modocal_bf: out std_logic_vector(7 downto 0);
60           RG_modofiltro_bf: out std_logic_vector(7 downto 0);
61           -----
62     );
63 end registros;

```

FIGURA 107 : DECLARACIÓN DE ENTIDAD DE "REGISTROS"

5.5.6. INTERFAZ PUERTO SERIE

En cuanto al módulo que dará el comportamiento del interfaz del puerto serie, se trata de un interfaz que se encuentra en las librerías de módulos reutilizables de INDRA SISTEMAS. La descomposición jerárquica del mismo es:

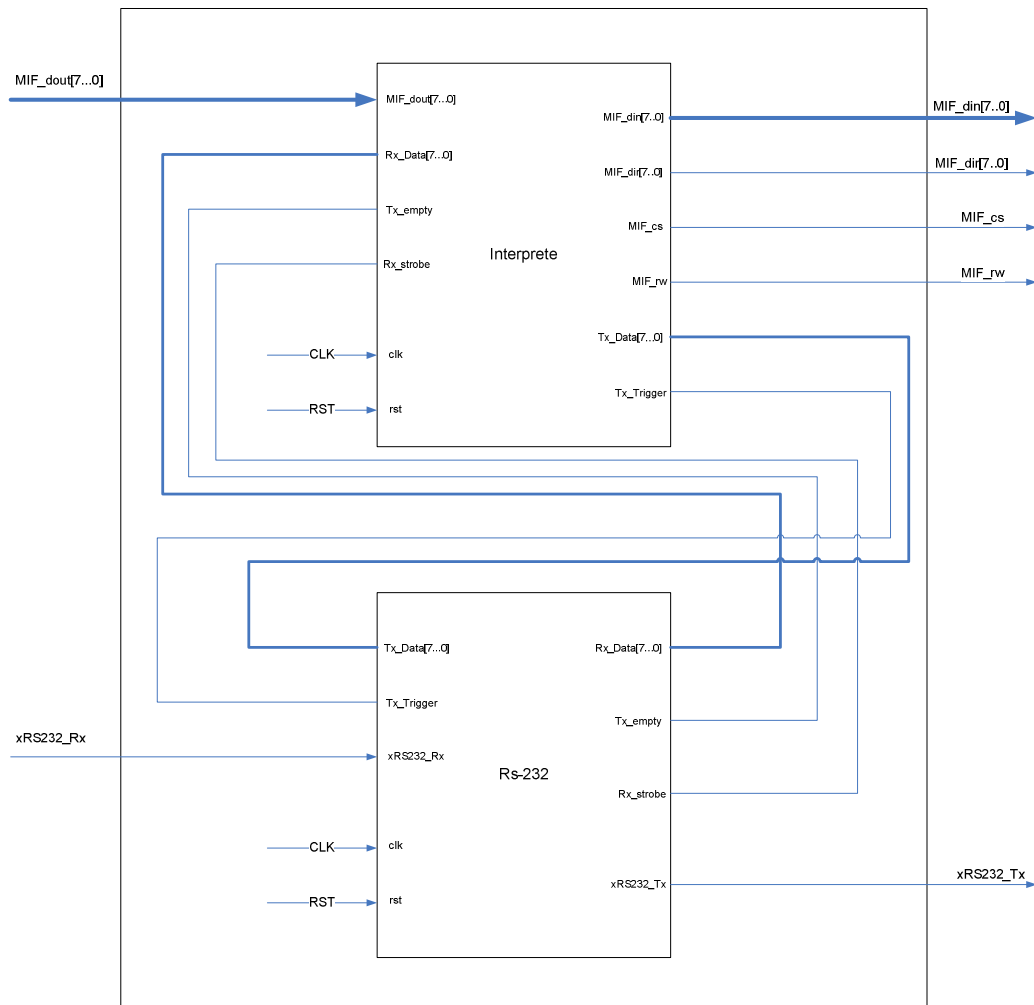


FIGURA 108: BLOQUE INTERFAZ RS-232

La declaración de entidad de este modulo es la siguiente:

```

20 library ieee;
21 use ieee.std_logic_1164.all;
22 use ieee.std_logic_unsigned.all;
23 use ieee.std_logic_arith.all;
24
25 --library unisim;
26 --use unisim.vcomponents.all;
27 --library virtex;
28 --use virtex.components.all;
29
30 entity RS232_Interfaz is
31 port(
32   --p- RS232 Interface, external pin
33   xRS232_Rx      : in std_logic;
34   xRS232_Tx      : out std_logic;
35   --p- Memory Interface
36   MIF_dout       : in std_logic_vector( 7 downto 0);
37   MIF_din        : out std_logic_vector( 7 downto 0);
38   MIF_dir        : out std_logic_vector(15 downto 0);
39   MIF_rw         : out std_logic;
40   MIF_cs         : out std_logic;
41   --p- .
42   clk            : in std_logic;
43   reset          : in std_logic
44 );
45 end RS232_Interfaz;
```

FIGURA 109 : DECLARACIÓN DE ENTIDAD DE "RS-232_INTERFAZ"

Y el cuerpo de arquitectura será el siguiente:

```

47 architecture RS232_Interfaz_arch of RS232_Interfaz is
48 --p- Transmitter Protocol Interface
49 signal Tx_Data      :std_logic_vector(7 downto 0);
50 signal Tx_Trigger    :std_logic;
51 signal Tx_Empty      :std_logic;
52
53 --p- Receiver Protocol Interface
54 signal Rx_Data       :std_logic_vector(7 downto 0);
55 signal Rx_Strobe_tr   :std_logic;
56
57 begin
58
59     urs232: entity work.RS232 (RS232_arch)
60     port map (
61         --p- RS232 Interface, external pin
62         xRS232_Rx      => xRS232_Rx      , --: in std_logic;
63         xRS232_Tx      => xRS232_Tx      , --:out std_logic;
64         --p- Transmitter Protocol Interface
65         Tx_Data        => Tx_Data        , --: in std_logic_vector(7 downto 0);
66         Tx_Trigger      => Tx_Trigger      , --: in std_logic;
67         Tx_Empty        => Tx_Empty        , --:out std_logic;
68         --p- Receiver Protocol Interface
69         Rx_Data        => Rx_Data        , --:out std_logic_vector(7 downto 0);
70         Rx_Strobe_tr    => Rx_Strobe_tr    , --:out std_logic;
71         --p- .
72         clk            => clk            , --: in std_logic;
73         reset          => reset          , --: in std_logic
74     );
75
76     uinter: entity work.RS232_Interprete(RS232_Interprete_arch)
77     port map (
78         --p- Transmitter Protocol Interface
79         Tx_Data        => Tx_Data        , --:out std_logic_vector(7 downto 0);
80         Tx_Trigger      => Tx_Trigger      , --:out std_logic;
81         Tx_Empty        => Tx_Empty        , --: in std_logic;
82         --p- Receiver Protocol Interface
83         Rx_Data        => Rx_Data        , --: in std_logic_vector(7 downto 0);
84         Rx_Strobe_tr    => Rx_Strobe_tr    , --: in std_logic;
85         --p- Memory Interface
86         MIF_dout        => MIF_dout        , --: in std_logic_vector( 7 downto 0);
87         MIF_din         => MIF_din         , --:out std_logic_vector( 7 downto 0);
88         MIF_dir         => MIF_dir         , --:out std_logic_vector(15 downto 0);
89         MIF_rw          => MIF_rw          , --:out std_logic;
90         MIF_cs          => MIF_cs          , --:out std_logic;
91         --p- .
92         clk            => clk            , --: in std_logic;
93         reset          => reset          , --: in std_logic
94     );
95
96 end RS232_Interfaz_arch;

```

FIGURA 110 : CUERPO DE ARQUITECTURA DE "RS-232_INTERFAZ"

Reutilizar bloques de código y almacenarlos en librerías resulta una práctica muy útil puesto que tan sólo habrá que añadir los ficheros fuentes al proyecto ya que se trata de código depurado y probado en otros sistemas.

5.5.7. FICHERO DE MAYOR NIVEL. TOP.VHD

Una vez implementados todos los bloques funcionales, se puede construir ya el fichero de alto nivel que contiene el comportamiento final de la FPGA que se encuentra en la TIC del EAE, objeto de este Proyecto Fin de Carrera.

Se trata de un sistema digital que es modelado por ese conjunto de módulos interconectados llamados bloques funcionales. En la siguiente ventana se observa como quedan finalmente instanciados todos los módulos anteriores en el proyecto de ISE Project Navigator:

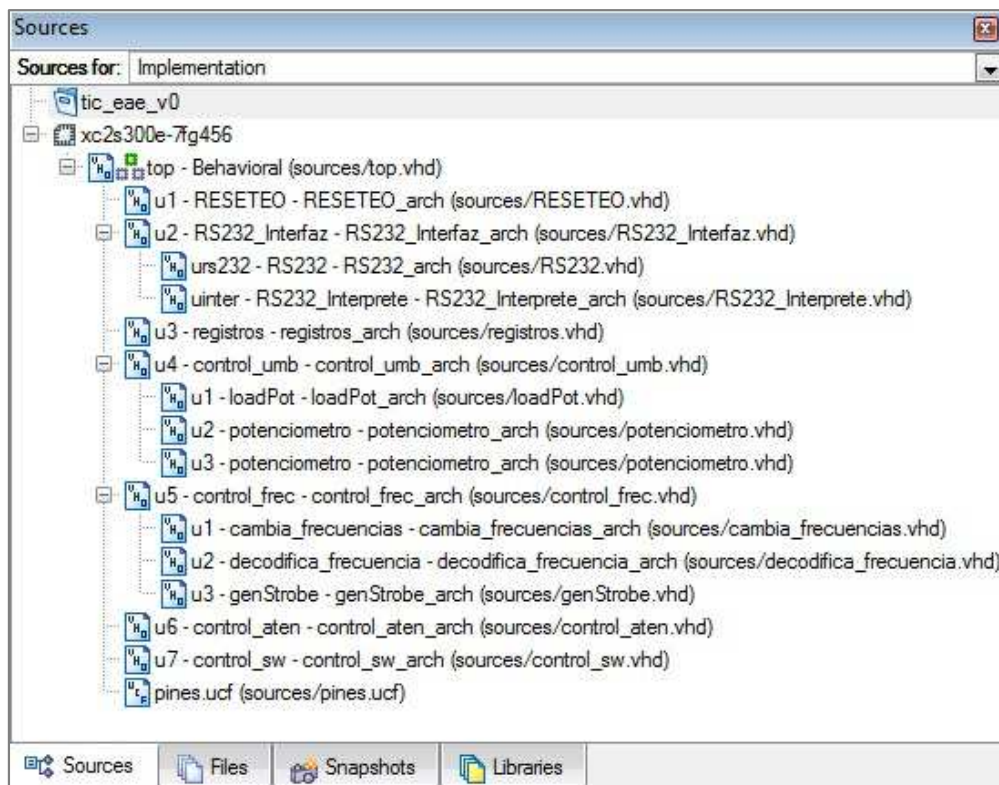


FIGURA 111 : VENTANA SOURCES DEL PROYECTO "TIC_EAE"

En la siguiente figura se observa un gran bloque en forma de caja negra del cual entran y salen señales. Se trata del fichero de más alto nivel de la jerarquía.

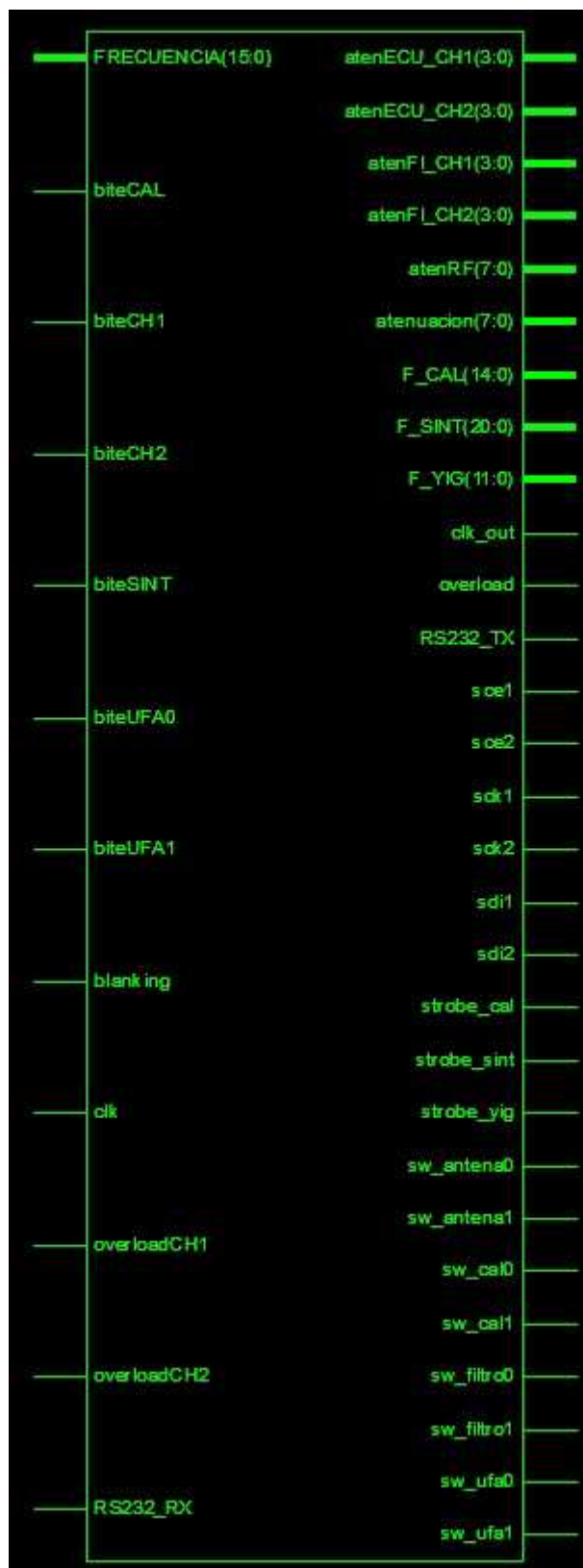


FIGURA 112 : ESQUEMA RTL OBTENIDO DEL FICHERO "TOP.VHD"

A continuación se muestra su estructura interior:

5.6. COMPLEJIDAD DEL DISEÑO.

Cuando se trata de diseños sobre FPGAs, el área de trabajo de ISE Project Navigator presenta una ventana denominada “Design Summary” que enumera la siguiente información:

- Información general, incluyendo el nombre del proyecto, tipo de dispositivo y el día y la fecha que se generó ese resumen.
- Resumen de utilización de dispositivos, que muestra una estimación del uso después de la síntesis y el uso después de realizar el mapeado.
- Datos de rendimiento generado en el informe después del “Place&Route”
- Información de restricciones, incluyendo restricciones de tiempos.
- Resumen de todos los informes con enlaces a ellos.

Por defecto, la ventana de “Design Summary” aparece en el área de Trabajo nada más abrir un proyecto, pero si no es así se puede acceder mediante *Project*→ *View Design Summary*, y el aspecto que muestra es el siguiente:

tic_eae Project Status					
Project File:	tic_eae.isc	Current State:	Placed and Routed		
Module Name:	top	• Errors:	No Errors		
Target Device:	xc2s300e-7fg456	• Warnings:	46 Warnings		
Product Version:	ISE 10.1.03 - Foundation	• Routing Results:	All Signals Completely Routed		
Design Goal:	Balanced	• Timing Constraints:	All Constraints Met		
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:	0 (Timing Report)		

tic_eae Partition Summary					
No partition information was found.					

Device Utilization Summary					
Logic Utilization	Used	Available	Utilization	Note(s)	
Number of Slice Flip Flops	499	6,144	8%		
Number of 4 input LUTs	861	6,144	14%		
Logic Distribution					
Number of occupied Slices	642	3,072	20%		
Number of Slices containing only related logic	642	642	100%		
Number of Slices containing unrelated logic	0	642	0%		
Total Number of 4 input LUTs	907	6,144	14%		
Number used as logic	861				
Number used as a route-thru	38				
Number used as Shift registers	8				
Number of bonded IOBs					
Number of bonded	126	325	38%		
Number of GCLKs	1	4	25%		
Number of GCLKIOBs	1	4	25%		

Performance Summary				
Final Timing Score:	0	Pinout Data:	Pinout Report	
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report	
Timing Constraints:	All Constraints Met			

Detailed Reports						
Report Name	Status	Generated	Errors	Warnings	Infos	
Synthesis Report	Current	dom 9. mar 14:47:49 2014	0	45 Warnings	18 Infos	
Translation Report	Current	dom 9. mar 14:53:38 2014	0	0	0	
Map Report	Current	dom 9. mar 14:53:47 2014	0	1 Warning	2 Infos	
Place and Route Report	Current	dom 9. mar 14:54:08 2014	0	0	2 Infos	
Static Timing Report	Current	dom 9. mar 14:54:14 2014	0	0	3 Infos	
Bitgen Report	Out of Date	sáb 8. mar 23:22:30 2014	0	0	0	

Date Generated: 07/02/2014 - 07:35:27

FIGURA 114 : VENTANA "DEVICE SUMMARY"

5.7. VALIDACIÓN FUNCIONAL. SIMULACIONES

Tras haber llevado a cabo la implementación del código y haber verificado que el modelo cumple las reglas semánticas y sintácticas, se lleva a cabo la simulación de dicho modelo para la comprobación de su correcto funcionamiento. Este apartado se ocupa de describir este paso en el flujo del diseño. Cuando las simulaciones resulten satisfactorias se llevarán a cabo las siguientes de etapas del flujo de diseño.

En VHDL simular consiste en conectar una serie de estímulos a las entradas del módulo implementado y verificar que las salidas adquieren el comportamiento final adecuado. Esto se realiza mediante la construcción de un fichero que contiene el código correspondiente a la composición de esos estímulos de entrada. Este fichero fuente se denomina “test bench” y contiene, igual que el modelo al que pone a prueba, una Declaración de Entidad y un Cuerpo de Arquitectura aunque con algunas diferencias.

Estos ficheros fuentes se añaden al proyecto de ISE a través de “*Project→New Source*”, y en el cuadro de diálogo que aparece se debe señalar que es un fichero fuente tipo “VHDL Test Bench”.

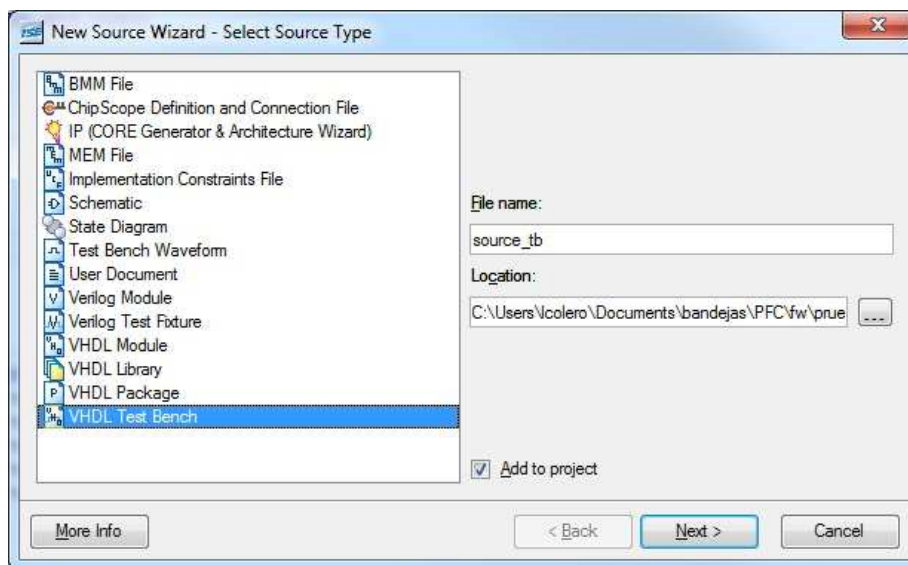


FIGURA 115 : VENTANA SELECCIÓN FICHERO DE SIMULACIÓN

A continuación, en el mismo cuadro de diálogo, se puede seleccionar qué modelo es el que se desea simular, y a partir de esto ISE Project Navigator construye un “esqueleto” del fichero test bench como el ejemplo siguiente:

```

27 -----
28 LIBRARY ieee;
29 USE ieee.std_logic_1164.ALL;
30 USE ieee.std_logic_unsigned.all;
31 USE ieee.numeric_std.ALL;
32
33 ENTITY source_tb IS
34 END source_tb;
35
36 ARCHITECTURE behavior OF source_tb IS
37
38     -- Component Declaration for the Unit Under Test (UUT)
39
40     COMPONENT source
41     PORT (
42         input : IN  std_logic;
43         output: OUT std_logic;
44     );
45     END COMPONENT;
46
47     --Inputs
48     signal input : std_logic := '0';
49     --Outputs
50     signal output : std_logic;
51
52     -- Clock period definitions
53     constant clk_period : time := 1us;
54
55 BEGIN
56     -- Instantiate the Unit Under Test (UUT)
57     uut: source PORT MAP (
58         input  => input,
59         output => output,
60     );
61
62     -- Clock process definitions
63     clk_process : process
64     begin
65         clk <= '0';
66         wait for clk_period/2;
67         clk <= '1';
68         wait for clk_period/2;
69     end process;
70
71     -- Stimulus process
72     stim_proc: process
73     begin
74         -- hold reset state for 100ms.
75         wait for 100ms;
76
77         wait for clk_period*10;
78
79         -- insert stimulus here
80
81         wait;
82     end process;
83
84 END;

```

FIGURA 116 : ESTRUCTURA DE UN "TEST_BENCH"

Para simular se empleará la herramienta “Xilinx ISE Simulator” del propio ISE Navigator, con el que se editará el “test bench” y se verificará el funcionamiento mediante el análisis de las formas de onda que se generan. La ventana de procesos para simular tendrá el aspecto como el del siguiente ejemplo:

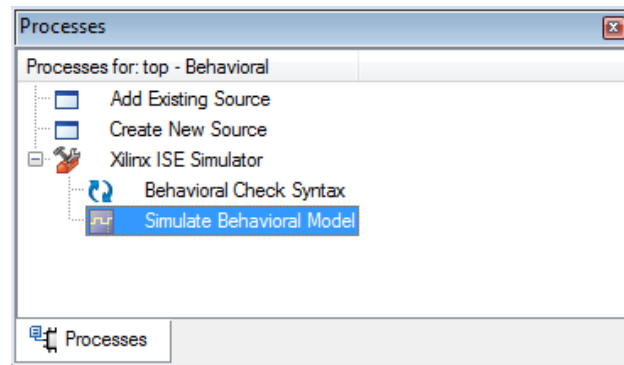


FIGURA 117 : VENTANA DE PROCESOS PARA SIMULACIÓN

A continuación se explicarán las principales simulaciones funcionales que se han llevado a cabo en este Proyecto Fin de Carrera.

5.7.1. BLOQUE “REGISTROS.VHD”

Para verificar el funcionamiento de este bloque, es necesario verificar que se realiza la escritura en los registros habilitados para ello, y poder realizar la lectura posteriormente. Para ello se manejarán los estímulos de entrada:

```
--inputs
MIF_din: in std_logic_vector (7 downto 0);
MIF_dir: in std_logic_vector (15 downto 0);
MIF_rw: in std_logic;
MIF_cs: in std_logic;
clk: in std_logic;
rst: in std_logic;
```

Se realizan varias simulaciones cuyos ficheros fuente son los siguientes:

- “registros_rd_tb.vhd”: indicando que se va a generar una lectura y la dirección del registro a leer, se lee el valor de cada registro de la tabla 37. Es importante que los valores de los registros estén según las especificaciones en el arranque para que el sistema pueda operar adecuadamente, por ello en esta simulación se pretende observar y verificar que se están cargando los valores por arranque y que se produce la lectura correctamente. Seguidamente se leen los valores de los registros de solo lectura, variando sus estímulos de entrada.
- “registros_wr_tb.vhd”: en este test bench se realiza la escritura y posterior lectura de cada registro habilitado para ello.

5.7.2. BLOQUE “CONTROL_UMB.VHD”

Los estímulos de entrada para este módulo son los siguientes:

```
--inputs
clk:    in  std_logic;
rst:    in  std_logic;
sdi_bf: in  std_logic_vector (7 downto 0);
carga:  in  std_logic;
```

El test bench que simula el funcionamiento de dicho módulo es el siguiente:

- `potenciometro_tb`: se debe indicar ciertos valores a cargar y una señal de indicación de carga para el mismo. Se debe verificar que se genera la frecuencia de reloj adecuada que necesitará para el dispositivo (AD5290) y que se suceden a partir de la señal de carga el resto de señales adecuadamente (SDI, SCK y SCE). Para ello se realiza la carga de varios valores.

5.7.3. BLOQUE “CONTROL_ATEN.VHD”

Los estímulos de entrada para este módulo son los siguientes:

```
--inputs
clk:    in  std_logic;
rst:    in  std_logic;
RG_atenFI_bf: in  std_logic_vector (7 downto 0);
RG_atenECU_bf: in  std_logic_vector (7 downto 0);
RG_atenRF_bf: in  std_logic_vector (7 downto 0);
```

El test bench que simula el funcionamiento de dicho módulo es el siguiente:

- `“control_aten_tb.vhd”`: se coloca a la entrada diversos valores de atenuación FI, ECU y RF, y se verifica que se distribuye correctamente a cada salida y que se obtiene un valor correcto de atenuación total.

5.7.4. BLOQUE “CONTROL_SW.VHD”

Los estímulos de entrada para este módulo son los siguientes:

```
--inputs
clk    : in  std_logic;
reset  : in  std_logic;
modo_func  : in  std_logic_vector (7 downto 0);
modo_cal   : in  std_logic_vector (7 downto 0);
modo_filtro: in  std_logic_vector (7 downto 0);
blanking  : in  std_logic_vector (7 downto 0);
```

El test bench que simula el funcionamiento de dicho módulo es el siguiente:

- “control_sw_tb.vhd”: se modifican los estímulos de entrada para verificar que los valores que salen correctamente en las señales de los switches. Se alterna entre el modo de funcionamiento antena y calibración, y se alterna entre el modo de calibración del EAE o del ECM, finalmente se varía la señal de “blanking” para diferentes casos y se comprueban los valores programados para los switches.

5.7.5. MÓDULO “CONTROL_FRECUENCIAS.VHD”

Los estímulos de entrada son:

```
--inputs
clk      : in  std_logic;
reset    : in  std_logic;
frecuencia: in  std_logic_vector (15 downto 0);
```

Se han efectuado las simulaciones necesarias para verificar el adecuado funcionamiento del módulo, obteniéndose los valores de frecuencia correctos según se especificó. Por tratarse de datos estratégicos esta información no será mostrada.

5.7.6. MÓDULO “TOP.VHD”

Los estímulos de entrada para este módulo son los siguientes:

```
--inputs
clk      : in  std_logic;
RS232_RX : in  std_logic;
FRECUENCIA : in  std_logic_vector (15 downto 0);
biteCH1: in  std_logic;
biteCH2: in  std_logic;
overloadCH1: in  std_logic;
overloadCH2: in  std_logic;
biteUFA0: in  std_logic;
biteUFA1: in  std_logic;
biteSINT: in  std_logic;
biteCAL: in  std_logic;
blanking: in  std_logic;
```

Se realiza varias simulaciones cuyos ficheros fuente son los siguientes:

- “top_232_tb”: se verifica la comunicación puerto serie, se realiza la escritura y posterior lectura del registro de comprobación de comunicación.
- “top_umbrales_tb”: se realiza la escritura sobre los dos registros (2001 y 2002) que recogen el valor de umbral y se comprueba el interfaz SPI a la salida.
- “top_atenuaciones_tb”: mediante el puerto serie se programarán los atenuadores y se verificará que producen el valor adecuado en las salidas.

- “top_switches_tb”: mediante el puerto serie se programará el modo de funcionamiento. Se irá variando el modo, y se verificará que las salidas hacia los switches adoptan las posiciones correctamente.
- “top_bites_tb”: mediante el puerto serie se leerán los valores de las señales de autocomprobación. Se van variando los estímulos de entrada y realizando lecturas en cada caso.

6. VERIFICACIÓN.

6.1. PRUEBAS DE ACEPTACIÓN

En este apartado se realizarán las primeras pruebas una vez obtenida físicamente la PCB soporte de la TIC del EAE objeto de este Proyecto Fin de Carrera. En este punto se cuenta ya materializada la tarjeta con los componentes montados.



FIGURA 118 : IMAGEN DE LA TARJETA IMPLEMENTADA FÍSICAMENTE. CARA DE COMPONENTES

Se realizan las pruebas básicas de aceptación que concluyen que la tarjeta cumple su propósito. Se describen a continuación las condiciones y tareas que se han llevado a cabo para verificarla y validarla.

Tal y como se explico en el punto 2.2.3 la TIC debe trabajar correctamente en el margen de temperaturas de -25°C a $+50^{\circ}$ y así mismo debe soportar un rango de temperaturas de almacenamiento de -25°C a $+70^{\circ}\text{C}$. La PCB una vez ensamblada en el módulo de la UMA, pasará dichas pruebas ambientales por el grupo de test del departamento de Producción y posteriormente pruebas de calidad en el departamento de Inspección de Calidad. Sin embargo, para su puesta en marcha, pruebas de aceptación y primeras comprobaciones, la TIC por sí misma no necesita consideraciones especiales al respecto, debido a que todos sus componentes y materiales han sido elegidos atendiendo a los márgenes ambientales.

Todos los instrumentos de medida y herramientas usados durante las pruebas de aceptación tendrán conforme su estado de calibración.

La primera tarea que se lleva a cabo es la inspección física de la TIC, se comprueba que ha sido fabricada conforme a la documentación del producto e incluye correctamente las siguientes comprobaciones:

- Identificación y marcado
- Dimensiones y masa

- Elementos de instalación y montaje
- No presenta daños mecánicos
- Orientación de los conectores e identificación de componentes.
- Acabados superficiales y limpieza.

La siguiente comprobación consiste en las pruebas eléctricas. Las pruebas eléctricas asegurarán la conformidad con la documentación del producto, especialmente para la continuidad del encaminamiento de alimentaciones y señales de control y ausencia de cortocircuitos en los pines de alimentación. Estas primeras pruebas eléctricas han resultado satisfactorias y son las siguientes:

- Presencia de “no continuidad” entre cada señal de alimentación y masa.
- Presencia de “no continuidad” entre las propias señales de alimentación.

La TIC del EAE formará parte del conjunto ensamblado de la UMA. Por lo tanto antes de realizar dicho montaje se deben llevar a cabo ciertas verificaciones.

- Consumos adecuados a las especificaciones.
- Comprobación de entrada de alimentaciones y salida de alimentaciones reguladas correctamente.

Para ello, se utilizarán tarjetas tipo “prototype-board” conectadas a la PCB a testear. Son placas empleadas por el presente departamento donde se está desarrollando este Proyecto Fin de Carrera, con el fin de hacer ciertas comprobaciones iniciales al arranque de los primeros ejemplares de las PCBs prototipo, y con la ventaja de que su coste es prácticamente nulo. Estas tarjetas suelen ser reutilizables para diferentes proyectos, y consiste en el montaje de terminales “faston” donde conectar las alimentaciones directamente de las fuentes de alimentación del laboratorio a la tarjeta “prototype-board”. Como ya se ha mencionado, estas primeras pruebas han resultado satisfactorias con lo que se concluye que la fabricación de la PCB ha salido exitosa.

6.2. INTEGRACIÓN.

Al concluir el proceso de diseño y fabricación de la Tarjeta Interfaz y Control del EAE, ésta pasará a formar parte dentro del modulo de la UMA tal y como se indicó en las especificaciones del punto 2.2.1 (ver figuras 6 y 7). Por lo tanto, las pruebas de validación finales de la TIC del EAE se realizarán en el conjunto superior ensamblado, es decir como un elemento más dentro del conjunto de la UMA y bajo las pruebas de aceptación de la UMA.

La PCB soporte de la Tarjeta Interfaz y Control del Equipo de Apuntamiento en Elevación dentro de su ensamble final en la mecánica de la bandeja inferior muestra la siguiente forma:

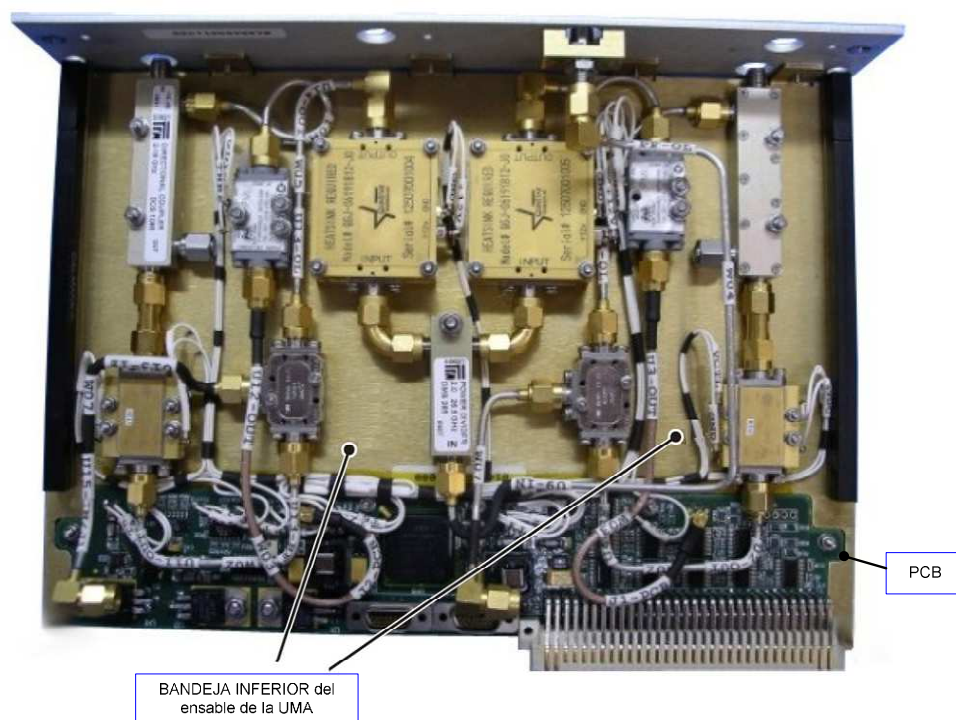


FIGURA 119 : VISTA DESPIECE BANDEJA INFERIOR

El aspecto del módulo ensamblado junto con la TIC es el siguiente:



FIGURA 120 : MÓDULO UMA VISTA SUPERIOR



FIGURA 121 : MÓDULO UMA VISTA FRONTAL

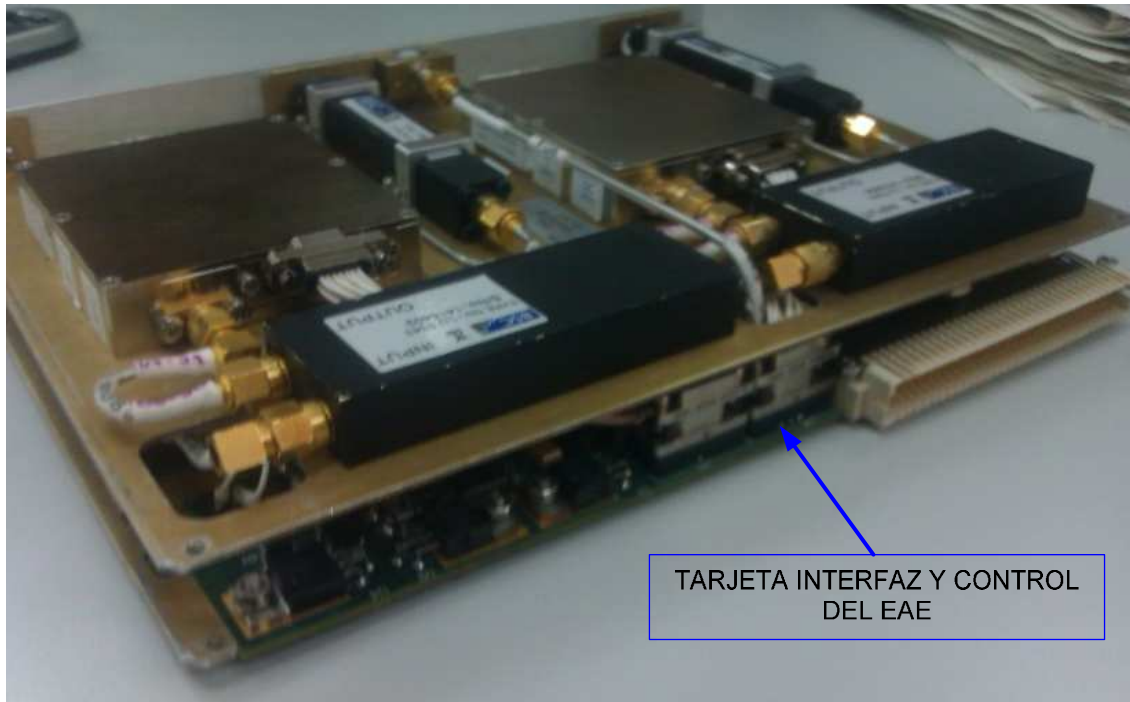


FIGURA 122 : MÓDULO UMA VISTA TRASERA

Es en este punto dónde se realizará la integración, en primer lugar con el módulo UMA donde pasará en conjunto pruebas de aceptación, validación funcional y de calidad. En segundo lugar, se realizará la integración con el conjunto de elementos con los que debe interactuar y finalmente en la mecánica del EAE con el resto de todos los elementos.

7. DOCUMENTACIÓN

Según la metodología de trabajo de INDRA SISTEMAS y, con el fin de poder definir y producir una tarjeta de forma inequívoca y precisa, ésta debe ir acompañada de un Paquete Documental. Toda la información que define una tarjeta debe quedar almacenada en las herramientas y aplicaciones determinadas para tal efecto. Además, es necesaria la asignación a la tarjeta del correspondiente código que la identificará.

Antes de continuar, es muy importante tener clara la distinción entre “PCB” y “Tarjeta”. El diseño de una tarjeta comienza con el diseño de un PCB, pero según los materiales que se monten éstos pueden otorgar diferentes funcionalidades y dar lugar a una tarjeta u otra. Por ello es necesaria la asignación del correspondiente código de tarjeta y código de PCB que los identifica respectivamente. Según la codificación, estos dos elementos son elementos distintos, siendo la PCB un código dependiente de la tarjeta a la que le ofrece soporte físico.

El Paquete Documental que se aplica a un producto determinado debe estar almacenado y disponible en el Gestor Documental. Ésta es una herramienta cuyo objetivo es permitir la búsqueda, edición y revisión de dicha documentación, que pasará por diferentes fases empezando por la fase de borrador y finalizando por la fase de firmas para su validación, momento en el cual se genera el Informe de Configuración Aplicable del producto en sí.

Por ello, durante el proceso de diseño de la Tarjeta de Interfaz y Control del EAE y después de su implementación, se ha llevado a cabo la elaboración de los documentos asociados que definen el producto. Ello debe tenerse en cuenta en la planificación de tareas del desarrollo del diseño.

De forma general, los documentos asociados sobre una tarjeta deben ser los siguientes:

- Esquema Eléctrico (EE): se trata de la representación gráfica del circuito en la que queda definido los componentes del circuito y la interconexión entre ellos. Refleja la situación real de la tarjeta montada tanto en valores de componentes montados como no montados.
- Lista de Materiales (LM): debe enumerar todos los materiales necesarios para el montaje. Los elementos de la LM deben mantener relación con los esquemas eléctricos mediante su referencia topológica. El PCB entra dentro de esta lista como un componente más de la tarjeta. Si hubiera algún elemento cuya huella aparece en el PCB pero no se monta, no debe aparecer en la lista de materiales.

- Plano de Montaje (PM): debe contener las referencias a todos los componentes electrónicos montados más la mecánica asociada al montaje de la misma (tornillería, etiquetas, etc...). También define todas las operaciones a realizar sobre el montaje (barnizado, pegado de componentes, cableados, etc...).
- Documento de Carga del FW (FI): debe describir la carga de todo el firmware. Este documento además define las versiones de firmware cargado en cada elemento programable. El firmware en sí mismo, debe estar contenido en un fichero comprimido que será referenciado en el documento de carga.
- Documento de Ajuste y Pruebas (AP): detalla las pruebas y ajustes necesarios para realizar la aceptación del producto y su puesta en marcha.

Si, después del cierre del Paquete Documental, ocurren modificaciones que atañen a algún elemento del paquete deberán quedar registradas. Cualquier propuesta de modificación a una tarjeta (cambio de firmware, cambio en algún documento asociado, en su lista de materiales,...) debe gestionarse a través de una Propuesta de Cambio de Ingeniería. En dicha propuesta se debe indicar que documentos están afectados por el cambio.

8. RESULTADOS Y CONCLUSIONES.

En este apartado se realiza una evaluación del proyecto valorando la consecución de los objetivos y la verificación del comportamiento final.

El objetivo principal del presente Proyecto Fin de Carrera consistía en aportar una solución a la necesidad de ampliar el número de líneas de control entre los diferentes módulos del Equipo de Apuntamiento en Elevación: entre la Tarjeta Procesadora y el Bloque de Recepción, entre la Tarjeta de Proceso y la Tarjeta Digitalizadora y entre el Bloque de Recepción y las Antenas Receptoras.

Para lograr este objetivo, se ha diseñado una placa de circuito impreso que contiene en ella una FPGA de Xilinx, así como el resto de elementos necesarios para cubrir la necesidad de aportar control y medio interfaz para las comunicaciones entre los módulos anteriormente mencionados.

De tal forma, en este PFC se ha desarrollado el diseño hardware de la tarjeta de circuito impreso soporte de la TIC, realizando previamente un análisis de los requisitos funcionales y no funcionales, analizando las especificaciones técnicas y valorando con qué materiales se da solución a todo ello. Ésta ha sido una de las tareas que más tiempo ha llevado, debido al volumen de especificaciones y limitaciones con las que se contaba

Posteriormente a este estudio, se han implementado los esquemas eléctricos de la placa de circuito impreso aportando la documentación de fabricación.

Así mismo, mientras se rutaba y fabricaba la PCB, se ha desarrollado el código VHDL que hace que la FPGA adquiera el comportamiento deseado para que la TIC realice las tareas descritas según las especificaciones dadas.

Una vez fabricada la PCB y montados los componentes en la misma, se ha logrado una Tarjeta Interfaz y de Control para los módulos del Equipo de Apuntamiento en Elevación. En este punto, con todos los componentes ya integrados, se puede ya manifestar que todo el diseño completo funciona correctamente.

A lo largo del desarrollo de este diseño se han ido emprendiendo las tareas que se habían definido previamente. Una vez abordadas y cerradas éstas se dan por concluidos los propósitos de este Proyecto Fin de Carrera.

En resumen las tareas que se han llevado a cabo en este Proyecto Fin de Carrera dentro del marco de trabajo de INDRA SISTEMAS S.A han sido

- Estudio funcional tanto del contexto operativo de la TIC dentro del EAE, así como el desglose de los bloques funcionales a desarrollar a partir de los requisitos.
- Estudio no funcional orientado a limitaciones de espacio, alimentaciones, consumos, etc... que conlleva el diseño de la TIC del EAE.
- Evaluar distintos trabajos acometidos con anterioridad dentro del departamento, buscando similitudes con las especificaciones de diseño y encontrando elementos reutilizables directa o parcialmente (componentes, librerías, estudios funcionales, ...)
- Planteamiento de soluciones, así como el análisis de arquitectura e interfaces entrada/salida de una FPGA de Xilinx.
- Creación de esquemáticos que describen la PCB soporte de la TIC, operando con la herramienta Design Capture de Mentor Graphics.
- Selección de componentes así como acopio de materiales, atendiendo al contexto de trabajo de INDRA SISTEMAS.
- Colaboración y soporte con otros departamentos que llevan a cabo el Ruteado, Fabricación y Montaje de la PCB que da soporte a la TIC del EAE.
- Diseño del firmware en VHDL operando con la herramienta ISE Navigator de Xilinx.
- Simulación funcional para la verificación del diseño firmware con la herramienta ISE Simulator de Xilinx.
- Revisión y Aceptación de la TIC, para su posterior integración.

Durante el proceso de desarrollo de este proyecto fin de carrera, la FPGA queda descatalogada. Por filosofía de la empresa, en este tipo de diseños es frecuente comprar suficientes unidades para disponer de ellas en stock. Por lo tanto, la empresa contaría con suficientes unidades por si fuera necesario reproducir la tarjeta o reparar alguna avería. No obstante se plantea ya en este mismo momento, un análisis de posibles alternativas.

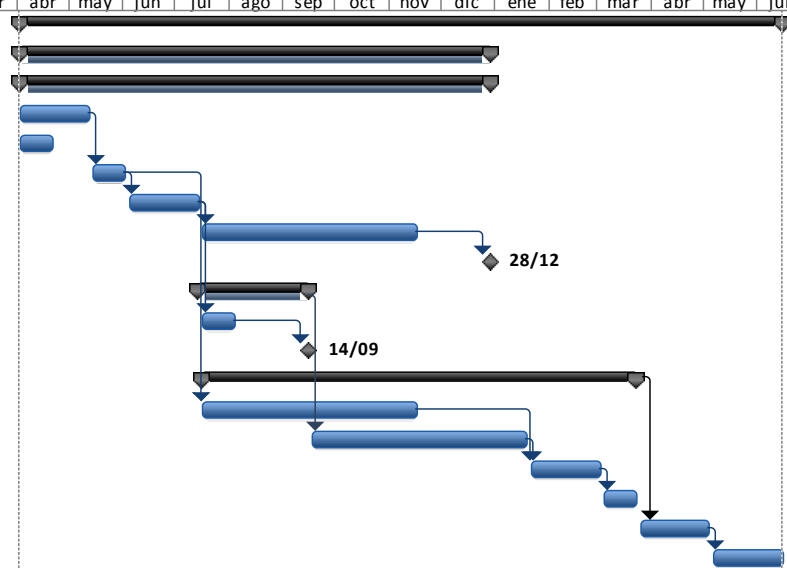
El lenguaje es independiente de la tecnología. El mismo modelo puede ser sintetizado en librerías de distintos vendedores. Por lo tanto ya que el firmware que se configura en la FPGA realizado en código VHDL es completamente reutilizable, así como el resto del diseño hardware de la tarjeta, se plantea para el futuro un rediseño con una alternativa para la FPGA.

9. PLANIFICACIÓN Y PRESUPUESTO.

9.1. PLANIFICACIÓN

Finalmente, se explica la programación que se ha llevado a cabo, y se indican las actividades comprendidas en el desarrollo de este proyecto, detallando paquetes de trabajo y tareas:

Id	Nombre de tarea	Duración	Comienzo	Fin																		
					tri 2, 2012			tri 3, 2012			tri 4, 2012			tri 1, 2013			tri 2, 2013			tri 3, 2013		
					mar	abr	may	jun	jul	ago	sep	oct	nov	dic	ene	feb	mar	abr	may	jun	jul	ago
1	B - PROYECTO	315 días	lun 02/04/12	vie 14/06/13																		
2	PT1 - DISEÑO	195 días	lun 02/04/12	vie 28/12/12																		
3	T1 - INGENIERIA	195 días	lun 02/04/12	vie 28/12/12																		
4	T1.1 - ANALISIS DE REQUISITOS	30 días	lun 02/04/12	vie 11/05/12																		
5	T1.2 - DIAGRAMA DE BLOQUES	15 días	lun 02/04/12	vie 20/04/12																		
6	T1.3 - SELECCIÓN DE COMPONENTES	15 días	lun 14/05/12	vie 01/06/12																		
7	T1.4 - ESQUEMAS ELECTRICOS	30 días	lun 04/06/12	vie 13/07/12																		
8	T1.5 - FIRMWARE - VHDL	90 días	lun 16/07/12	vie 16/11/12																		
9	T1.6 - DOCUMENTACION	30 días	lun 19/11/12	vie 28/12/12																		
10	T2 - PCB	46 días	vie 13/07/12	vie 14/09/12																		
11	T2.1 - ESPECIFICACION PCB	15 días	lun 16/07/12	vie 03/08/12																		
12	T2.2 - DOCUMENTACION	30 días	lun 06/08/12	vie 14/09/12																		
13	PT2 - FABRICACION Y PRUEBAS	180 días	lun 16/07/12	vie 22/03/13																		
14	T1 - ACOPIO DE MATERIALES	90 días	lun 16/07/12	vie 16/11/12																		
15	T2 - FABRICACION PCB	90 días	lun 17/09/12	vie 18/01/13																		
16	T3 - MONTAJE	30 días	lun 21/01/13	vie 01/03/13																		
17	T4 - PRUEBAS	15 días	lun 04/03/13	vie 22/03/13																		
18	PT3 - INTEGRACION	30 días	lun 25/03/13	vie 03/05/13																		
19	PT4 - CIERRE PAQUETE DOCUMENTAL	30 días	lun 06/05/13	vie 14/06/13																		



9.2. COSTES GENERALES

El coste del proyecto se presupuesta recogiendo las tareas detalladas en el punto anterior y organizando los costes según la tabla siguiente:

CONCEPTO	RECURRENTE			NO RECURRENTE			TOTAL
	HORAS	€/H	TOTAL (€)	HORAS	€/H	TOTAL (€)	
Diseño (Ingeniero. Desarrollo)				200	25,00	5000,00	5000,00
Documentación				20	25,00	500,00	500,00
Diseño Esquema PCB				40	25,00	1000,00	1000,00
Diseño Ruteado PCB				40	25,00	1000,00	1000,00
Fabricación PCB	20	-	120,00				120,00
Gestión Acopio Materiales	1	25,00	25,00				25,00
Compra Materiales	-	-	500,00				500,00
Almacén (movimiento de materiales)	3	12,00	36,00				36,00
Mano de Obra de Montaje	30	20,00	600,00				600,00
Inspección de Calidad	1	18,25	18,25				18,25
Test	1	18,25	18,25				18,25
Integración	120	25,00	3125,00				3125,00

TOTAL:	11.942,5 €
--------	------------

9.3. ANÁLISIS DE OBSOLESCENCIA FPGA.

Como se menciona en el apartado anterior “Resultados y Conclusiones” una vez realizado el desarrollo del HW, del Firmware en VHDL, la puesta en marcha, pruebas, etc... se ha detectado por comunicación del proveedor, que la FPGA utilizada entra en fase de EOL (End Of Live) lo cual implica tener que realizar un análisis de alternativas al producto, que se debe tener fundamentalmente en cuenta en cuanto a costes se refiere.

➤ **Alternativa 1:** Mediante el acopio de materiales.

Para realizar el mantenimiento durante el ciclo de vida del sistema, se puede optar por realizar un acopio de materiales suficiente como para mantener el HW operativo, tanto por averías como para fabricaciones nuevas por mantenimiento.

Las consideraciones a tener en cuenta son principalmente tres: unidades fabricadas, unidades previstas a fabricar y tiempo esperado de mantenimiento.

- Unidades fabricadas : 4
- Unidades a fabricar : 2
- Tiempo de Mantenimiento: 15

Considerando un MTBF (Mean Time Between Failure) en torno a 7.500,00 horas, y FUA (factor de utilización anual) de una media de 4h/día durante 365 días/año, serían 1460h/año.

En función de su MTBF y su FUA se podría dictaminar que cada 5 años aproximadamente, se averiaría la tarjeta; pensando de forma pesimista y que la avería fuera irreparable, lo cual implicase el fabricar una tarjeta nueva. Esto quiere decir que en 15 años se averiarían 3 tarjetas y debería fabricarse entonces 3 unidades nuevas por cada unidad fabricada, es decir

- Unidades fabricadas : 4 ==> fabricar 12
- Unidades a fabricar : 2 ==> fabricar 6

En 15 años sería fabricar una media de 1 por cada año a partir del quinto año.

Lo cual para mantener 6 sistemas, durante los próximos 15 años, se podría realizar un acopio de mínimo 18 uds de FGPA's para dar solución al ciclo de vida, pero tomando un 10% de margen de error por otras averías, bien podría añadirles 1.8 uds más, es decir en números redondos unas 20uds.

Para el caso de la FPGA XC2S300E, y considerando un coste unitario en torno a los 30€, sería una inversión de 600€, más el coste de almacenamiento.

➤ **Alternativa 2:** Actualización del Diseño

Significa abordar un nuevo diseño, que contemple el cambiar el HW obsoleto de la FPGA por una nueva que tenga una esperanza de vida en torno a 10 años.

Para ello a continuación se establecen las siguientes tareas a realizar:

- Tarea 1 (T1) - Búsqueda de alternativa HW: consiste en investigar una FPGA equivalente en coste, en prestaciones, en tamaño y capacidades. Se realizarán búsquedas comparativas por internet y en algún caso hablar con representantes comerciales. Se estima que esta tarea puede ocupar en torno a 1 semana de trabajo. (40h).
- Tarea 2 (T2) - Rediseño del HW: una vez elegida la alternativa, se procederá a reeditar los esquemáticos existentes y sustituir los componentes obsoletos por la nueva configuración, siempre intentando mantener lo máximo existente del circuito y aprovechar el “know how”. Se estima que esta tarea puede ocupar en torno a 2 semanas de trabajo (80h).
 - Tarea 2.1 (T2.1) - Rediseño del PCB: surge como trabajo derivado de la tarea T2, dado que los cambios en el HW afectan directamente al ruteado de las pistas. Se estima que esta tarea puede ocupar en torno a 1 semana de trabajo (40h) más una subcontratación del ruteado.
- Tarea 3 (T3) - Rediseño del FW: al haber un cambio de FPGA, como mínimo implicaría cambios en el fichero de configuración de entradas y salidas (fichero “pines.ucf”), posteriormente recompilar y generar un fichero a cargar en el nuevo diseño. Se estima que esta tarea puede ocupar en torno a 1 semanas de trabajo (40h).
- Tarea 4 (T4) - Fabricación: se fabricaría un prototipo para validar el rediseño. Se estima que esta tarea puede ocupar en torno a 3 - 4 meses, debido a acopios de materiales, pero no un 100% de dedicación en horas por personas, se podría contar 1 semana por mes (160h) además de una subcontratación de montajes por la complejidad de la circuitería.

- Tarea 5 (T5) - Pruebas: consiste en una vez rediseñado y fabricado un prototipo, comprobar que los cambios son transparentes a la funcionalidad ya existente. Se cargaría el nuevo FW descrito en el T3. Se estima que esta tarea puede ocupar en torno a 2 semanas de trabajo (80h).
- T6 - Documentación: consiste en documentar todos los cambios tales como esquemáticos, listas de materiales, configuración, pruebas, etc... resultantes de todos los trabajos realizados en las tareas descritas antes. Se estima que esta tarea puede ocupar en torno a 2 semanas de trabajo (80h).

El resumen de costes sería según la tabla siguiente:

TASA MEDIA(€) = 25

TAREA	DESCRIPCION	MANO DE OBRA (H)	SUBCONTRATACIÓN (€)	MATERIALES (€)
T1	Búsqueda alternativa HW	40	0,00	0,00
T2	Rediseño HW	80	0,00	0,00
T2.1	Rediseño del PCB	40	1500,00	0,00
T3	Rediseño del FW	40	0,00	0,00
T4	Fabricación	160	600,00	1500,00
T5	Pruebas	80	0,00	0,00
T6	Documentación	80	0,00	0,00
		14000,00	2100,00	1500,00
				Total 17600,00

Por último sería necesario, realizar una planificación de estas nuevas tareas como trabajos futuros. El cronograma de las tareas será como el que se muestra en la siguiente figura:

Id	Nombre de tarea	Duración	Comienzo	Fin	Predec	Gantt Chart Timeline																											
						sep '13					oct '13					nov '13					dic '13					ene '14					feb '14		
						26	02	09	16	23	30	07	14	21	28	04	11	18	25	02	09	16	23	30	06	13	20	27	03	10			
1	Solución Obsolescencia	110 días	lun 02/09/13	vie 31/01/14																													
2	T1 - Búsqueda alternativa HW	5 días	lun 02/09/13	vie 06/09/13																													
3	T2 - Rediseño HW	15 días	lun 09/09/13	vie 27/09/13	2																												
4	T2.0. Rediseño HW	10 días	lun 09/09/13	vie 20/09/13																													
5	T2.1 -Rediseño de PCB	5 días	lun 23/09/13	vie 27/09/13	4																												
6	T3 - Rediseño FW	80 días	lun 23/09/13	vie 10/01/14	4																												
7	T4 - Fabricación	10 días	lun 30/09/13	vie 11/10/13	5																												
8	T5 - Pruebas	10 días	lun 13/01/14	vie 24/01/14	7;6																												
9	T6. Documentación	10 días	lun 30/09/13	vie 11/10/13	3																												

Después de este análisis, la conclusión que se obtiene es que la alternativa 1 es mucho más rentable económicamente que la alternativa 2, aunque esta última sea la solución más correcta. Pero se trata de ofrecer un producto final adecuado para las necesidades del cliente pero basándose en las posibilidades de la empresa para lograr rentabilidad.

10. ANEXOS

Los anexos de este Proyecto Fin de Carrera, que se han incluido en el CD, son los siguientes:

- ✓ ESQUEMAS ELÉCTRICOS: documento PDF que contiene los planos de la representación de los componentes y sus interconexiones.
- ✓ LISTA DE MATERIALES: documento PDF con la tabla que indica los materiales necesarios para el montaje de la TIC del EAE, así como número de unidades, partname, descripción, etc...
- ✓ DATASHEETs DE COMPONENTES: carpeta que contiene las hojas de características de los principales componentes que van montados en la TIC.
- ✓ INTERFACES ENTRADA/SALIDA : carpeta que contiene las tablas con la información de las señales para consulta rápida del lector:
 - Interfaz de E/S de la TIC: recoge las entradas y salidas de la TIC, indicando origen y destino.
 - Interfaz E/S de la FPGA: recoge la interfaz del encapsulado de la FPGA.
- ✓ CODIGOS FUENTE Y FICHEROS DE SIMULACIÓN: carpeta que incluye los códigos fuente del diseño en VHDL y los ficheros de simulación más significativos.
- ✓ DIAGRAMA DE GANTT: se incluirá los ficheros de Microsoft Project que definen la planificación de tareas.

11. GLOSARIO DE TÉRMINOS Y ABREVIATURAS

AH: Array de antenas combinadas en horizontal

AV: Array de antenas combinadas en vertical

ARX: Antenas receptoras

BITE: señal de autocomprobación (Built-in-test Equipment;)

BOM: Bill Of Materials

CPLD: Complex Programmable Logic Device

CYC Calibrado y Conversión

DAC: Convertidor analógico-digital (Digital analog converter)

DPC Bloque de Digitalización, Proceso y Control

EAE: Equipo de Apuntamiento en Elevación

ECU: Ecualización

ECM: Electronic Countermeasures

ESM: Electronic Support Measure

FI: Frecuencia Intermedia

FPGA: Field Programmable Gate Array

FW: Firmware

GND: masa o tierra

HW: Hardware

LVTTL: Low Voltage TTL. Niveles lógicos entre 0 y 3.3V

PCB: Tarjeta de circuito impreso (Printed circuit board)

PFC: Proyecto Fin de Carrera

PROM: Programmable Read-Only Memory

RF: Radio Frecuencia

RTL: Register Transfer Level

RX: recepción

SPI: Serial Peripheral Interface

SW: Software

TIC: Tarjeta Interfaz y de Control

TD: Tarjeta Digitalizadora o Tarjeta de Digitalización

TP: Tarjeta Procesadora o Tarjeta de Proceso

TTL: niveles lógicos entre 0 y 5V

TX: transmisión

UC: Unidad de Calibrado

UMA: Unidad de Mezcla y Adaptación

UFA: Unidad de Filtrado y Amplificación

UOL: Unidad de Oscilador Local

VHDL: VHSIC Hardware Description Language

12. BIBLIOGRAFÍA

[1]Xilinx, inc.: Spartan IIE 2.5VFPGA Family: Functional Description. In: Product Specification, disponible en <http://www.xilinx.com>.

[2]Using Select IO Interface in Spartan-II and Spartan IIE FPGAs
http://www.xilinx.com/support/documentation/application_notes/xapp179.pdf

[3]Xilinx Device Package User Guide, disponible en
http://www.xilinx.com/support/documentation/user_guides/ug112.pdf

[3]Desing Capture for Expedition PCB. Training Manual, disponible en:
http://www.ict.kth.se/courses/IL2208/ESP_IL2208_MentorGraphics/DesignCapture.pdf

[4]Desing Capture Desing Process Guide, disponible en:
<http://www.mentor.com/training/courses/design-capture-for-expedition-pcb-layout>

[5] Product Discontinuation Notice: Spartan-IIE, Virtez-E, Virtex-EM, Virtex-II, EasyPath Virtex-II and XC17V00 PROMs FPGA Products. (April 24, 2013), disponible en
http://www.xilinx.com/support/documentation/customer_notices/xcn12026.pdf]

PÁGINA EN BLANCO INTENCIONADAMENTE